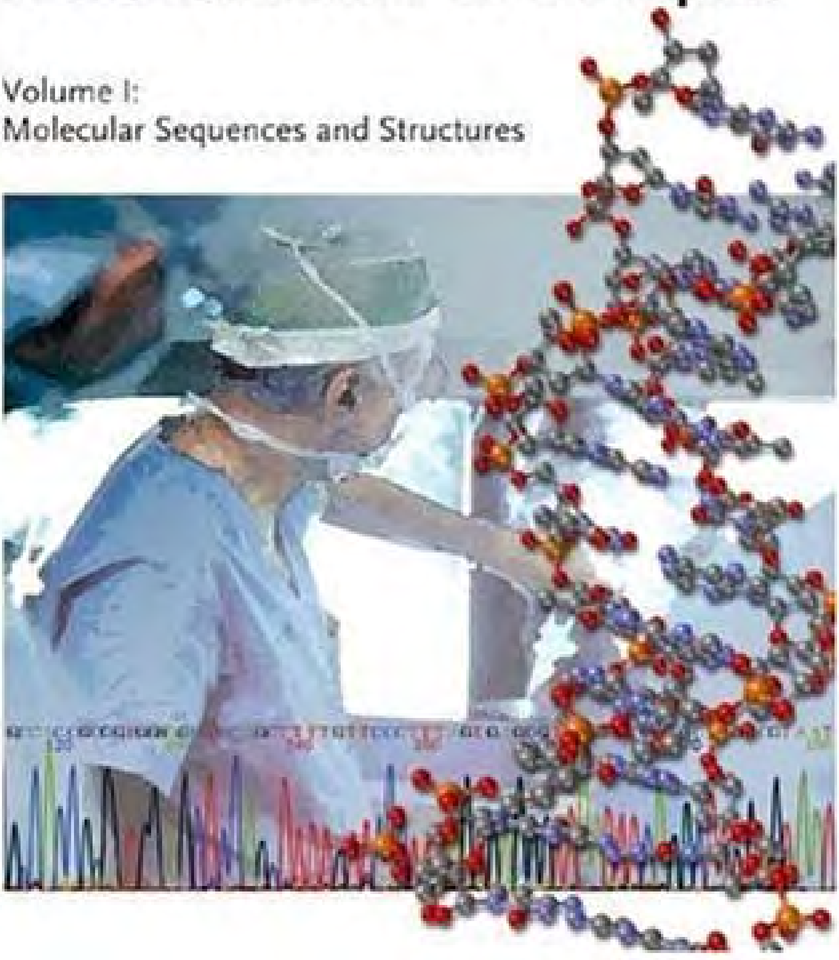
Edited by Thomas Lengauer

WILEY-VCH

Bioinformatics – From Genomes to Therapies

Volume I: Molecular Sequences and Structures



Bioinformatics –
From Genomes to Therapies

Edited by
Thomas Lengauer

Bioinformatics – From Genomes to Therapies Volume 1

The Building Blocks: Molecular Sequences and Structures

Edited by
Thomas Lengauer

WILEY-VCH Verlag GmbH & Co. KGaA

Bioinformatics – From Genomes to Therapies Volume 2

Getting at the Inner Workings: Molecular Interactions

Edited by
Thomas Lengauer

BICENTENNIAL
1807
WILEY
2007
BICENTENNIAL

WILEY-VCH Verlag GmbH & Co. KGaA

Bioinformatics –
From Genomes to Therapies
Volume 3

The Holy Grail:
Molecular Function

Edited by
Thomas Lengauer

WILEY-VCH Verlag GmbH & Co. KGaA

1807–2007 Knowledge for Generations

Each generation has its unique needs and aspirations. When Charles Wiley first opened his small printing shop in lower Manhattan in 1807, it was a generation of boundless potential searching for an identity. And we were there, helping to define a new American literary tradition. Over half a century later, in the midst of the Second Industrial Revolution, it was a generation focused on building the future. Once again, we were there, supplying the critical scientific, technical, and engineering knowledge that helped frame the world. Throughout the 20th Century, and into the new millennium, nations began to reach out beyond their own borders and a new international community was born. Wiley was there, expanding its operations around the world to enable a global exchange of ideas, opinions, and know-how.

For 200 years, Wiley has been an integral part of each generation's journey, enabling the flow of information and understanding necessary to meet their needs and fulfill their aspirations. Today, bold new technologies are changing the way we live and learn. Wiley will be there, providing you the must-have knowledge you need to imagine new worlds, new possibilities, and new opportunities.

Generations come and go, but you can always count on Wiley to provide you the knowledge you need, when and where you need it!

William J. Pesce

William J. Pesce
President and Chief Executive Officer

Peter Booth Wiley

Peter Booth Wiley
Chairman of the Board

The Editor

Prof. Dr. Thomas Lengauer
Max-Planck-Institute
for Informatics
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

All books published by Wiley-VCH are carefully produced. Nevertheless, editors, authors and publisher do not warrant the information contained in these books to be free of errors. Readers are advised to keep in mind that statements, data, illustrations, procedural details or other items may inadvertently be inaccurate.

Library of Congress Card No.:
applied for

British Library Cataloguing-in-Publication Data:
A catalogue record for this book is available from the British Library.

Bibliographic information published by the Deutsche Nationalibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de

Printed in the Federal Republic of Germany
Printed on acid-free paper

Cover Design: Schulz Grafik-Design, Fussgönheim
Composition: Steingraeber Satztechnik GmbH, Ladenburg
Printing: betz-Druck GmbH, Darmstadt
Bookbinding: Litges & Dopf Buchbinderei GmbH, Heppenheim

ISBN: 978-3-527-31278-8

For Sybille, Sara and Nico

Contents

Volume 1

Bioinformatics – From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

Volume 2

Volume 3

Contents

Volume 1

Bioinformatics – From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

Volume 3

Contents

Volume 1

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

Volume 2

Volume 3

Preface

This book is a substantially expanded sequel to the book *Bioinformatics – From Genomes to Drugs* that appeared in 2002. Since the publication of the predecessor book the field of bioinformatics has experienced continuing and substantially accelerated growth in terms of the volume and diversity of available molecular data, as well as the development of methods for analyzing and interpreting these data. This book is a reflection of the dynamic maturation of the field. Like its predecessor, it discusses bioinformatics in the context of pharmaceutical and medical challenges pertaining to the understanding, diagnosis and therapy of diseases. The previous book covered bioinformatics issues accompanying the stages from the collection of genomic data across the elucidation of the molecular basis of disease and the identification of target proteins for drug design to the search for leads for potential drugs. This book extends this schema in various ways. First, the process line from genome to drug is extended downstream towards the optimization of drug leads and further towards the personalization of drug therapies, which is also beginning to be supported with bioinformatics methods. Second, the book covers the field in substantially more breadth. The different types of available data are discussed more comprehensively and in more detail. On the sequence side, two chapters on RNA have been added. The bioinformatics analysis of evolutionary relationships is addressed in several chapters. The discussion of protein structure has been significantly expanded. There are new sections on molecular networks, mRNA expression data and protein function, covering several chapters each. The disease-specific part of the book has also been expanded, including discussions of bacterial and viral infections. Finally, several chapters on informatics technologies employed for bioinformatics are included.

Bioinformatics is continuing to present one of the grand challenges of our times. It has a large basic research aspect, since we cannot claim to be close to understanding biological systems on an organism or even cellular level. At the same time, the field is faced with a strong demand for immediate solutions, because the genomic and postgenomic data that are being collected harbor

Bioinformatics - From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

many biological insights whose deciphering can be the basis for dramatic scientific and economical success, and is promising to have large impact on society.

The book is directed at readers who are interested in how bioinformatics can spur biological and medical innovation towards understanding, diagnosing and curing diseases. The book is designed to be useful to readers with a variety of backgrounds. Biologists, biochemists, pharmacologists, pharmacists and medical doctors can get an introduction into basic and practical issues of the computer-based part of handling and interpreting genomic, postgenomic and clinical data. In particular, many chapters point to bioinformatics software and data resources which are available on the Internet (often at no cost), and make an attempt at classifying and comparing those resources. For computer scientists and mathematicians, the book contains an introduction to the biological background and the necessary information in order to begin appreciating the difficulties and wonders of modeling complex biochemical and biomolecular issues by computer. Since the book caters to a readership with widely varying backgrounds, it also contains chapters with a diverse makeup. There are chapters that put the biology in the foreground and only sketch methodical issues, and a smaller number of chapters in which the algorithmic and statistical content dominates. By and large, the way in which the chapters are written reflects the viewpoint from which the authors, and that also often means the world-wide research community, approaches the respective topic.

The book contains a name and a subject index. A methodical index is integrated inside the subject index and points to those sections that present the master introductions to the quoted computational methods.

The world's leading experts have contributed their expertise and written largely autonomous chapters on the specific topics of this book. In order to render added coherence to the book, the chapters contain a large number of cross-references to aid in relating the topics of different chapters to each other. In a few cases, overlap between the chapters has been allowed to ensure the independent readability of the chapters.

I am grateful to the many people who helped make this book possible. Above all, I thank the 91 authors of contributed chapters who have shown extraordinary commitment during the draft and revision stages of their text. Ruth Christmann spent many hours helping me to master the logistic feat of collecting the texts, encouraging authors to keep to their commitments, handling versions and completing revisions with a special focus on reference lists. Joachim Büch kept the website for book authors alive and well maintained during the preparation and production process. Ray Loughlin did a superb job on copy-editing the book. Frank Weinreich and later Steffen Pauly were always responsive partners on the side of the publisher. Finally, I would like to express my deep gratitude to my wife Sybille and my children Sara and Nico who had to cope with my physical or mental absence too much while the project was ongoing. They gave the most for receiving the least.

Saarbrücken *Thomas Lengauer*
October 2006

List of Contributors

Bissan Al-Lazikani
Inpharmatica Ltd
1 New Oxford Street
London WC1A 1NU
UK

Russ B. Altman
Stanford University Medical Center
Department of Genetics
300 Pasteur Drive, Lane L301
Stanford, CA 94305-5120
USA

Ron D. Appel
Swiss Institute of Bioinformatics
CMU - 1, rue Michel Servet
1211 Geneva 4
Switzerland

Karl-Heinz Baringhaus
Sanofi-Aventis Deutschland GmbH
Chemical Science / Drug Design
65926 Frankfurt
Germany

Niko Beerenwinkel
Harvard University
Program for Evolutionary Dynamics
1 Brattle Square
Cambridge, MA 02138
USA

Asa Ben-Hur
Colorado State University
Department of Computer Science
222 University Services Center
601 South Howes Street
Fort Collins, CO 80523 USA
USA

Richard Bonneau
New York University
Department of Biology/Computer Science
Center for Comparative Functional Genomics
100 Washington Square East
New York, NY 10003-6688
USA

Philip E. Bourne
University of California-San Diego
Department of Pharmacology
9500 Gilman Drive
La Jolla, CA 92093-0505
USA

Guillaume Bourque
Genome Institute of Singapore
60 Biopolis Street, #02-01, Genome,
Singapore 138672
Singapore

Bioinformatics – From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

Yannick Brunner
University of Geneva
Biomedical Proteomics Research Group
Department of Structural Biology and Bioinformatics
Centre Médical Universitaire
1, rue Michel Servet
1211 Genève 4
Switzerland

Douglas Lee Brutlag
Stanford University
Department of Biochemistry
Beckman Center, B400, Mail Code 5307
Stanford, CA 94305-5307
USA

Richard R. Copley
University of Oxford
Wellcome Trust Centre for Human Genetics
Roosevelt Drive
Oxford OX3 7BN
UK

Martin Däumer
The University Hospital of Cologne
Institute for Virology
Fürst-Pückler-Str. 56
50935 Köln
Germany

Jörg Degen
University of Hamburg
Center for Bioinformatics Hamburg (ZBH)
Bundesstrasse 43
20146 Hamburg
Germany

Francisco S. Domingues
Max-Planck-Institute for Informatics
Computational Biology and Applied Algorithmics
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Roland L. Dunbrack, Jr.
Institute for Cancer Research
Fox Chase Cancer Center
333 Cottman Avenue
Philadelphia, PA 19111
USA

Ingo Ebersberger
Heinrich-Heine-University Düsseldorf
Bioinformatics
Universitätsstrasse 1, Geb. 25.02.02
40225 Düsseldorf
Germany

Kevin W. Eliceiri
University of Wisconsin
Laboratory for Optical and Computational Instrumentation
Department of Molecular Biology
1675 Observatory Drive
Madison, WI 53706
USA

Edward J. Feil
University of Bath
Department of Biology and Biochemistry
Bath BA2 7AY
UK

David Fell
Oxford Brookes University
School of Biological and Molecular Sciences
Headington Campus
Gipsy Lane
Oxford OX3 0BP
UK

Dawn Field
Centre for Ecology and Hydrology, Oxford
Molecular Evolution and Bioinformatics Section
Mansfield Road
Oxford OX1 3SR
UK

Anna Gaulton
Pfizer Global Research and Development
Pfizer Ltd
Ramsgate Road
Sandwich
Kent CT13 9NJ
UK

Adam Godzik
The Burnham Institute
10901 North Torrey Pines Road
La Jolla, CA 92037
USA

Ilya G. Goldberg
National Institute on Aging,
Gerontology Research Center
Image Informatics and Computational Biology Unit
Laboratory of Genetics
5600 Nathan Shock Drive
Baltimore, MD 21224-6825
USA

Johannes Goll
Forschungszentrum Karlsruhe
Institute for Toxicology and Genetics
Hermann-von-Helmholtz-Platz 1
76344 Eggenstein-Leopoldshafen
Germany

Brian J. Haas
The Institute for Genomic Research
9712 Medical Center Drive
Rockville, MD 20850
USA

Arndt von Haeseler
Center for Integrative Bioinformatics
Max F. Perutz Laboratories
Dr. Bohr Gasse 9
1030 Vienna
Austria

Anthony Hasseldine
Department of Pharmacology and Biological Chemistry
Mount Sinai School of Medicine
One Gustave L. Levy Place
New York, NY 10029
USA

Anja von Heydebreck
Max-Planck-Institute for Molecular Genetics
Ihnestrasse 63–73
14195 Berlin
Germany

Andreas Hildebrandt
Center for Bioinformatics
Building E11
P.O. Box 151150
66041 Saarbrücken
Germany

Harry Hochheiser
Towson University
Department of Computer and Information Sciences
7800 York Road, Room 425
Towson MD 21252
USA

Ivo L. Hofacker
University of Vienna
Institute for Theoretical Chemistry
Währingerstr. 17
1090 Vienna
Austria

Andrew Hopkins
Pfizer Global Research and Development
Pfizer Ltd
Ramsgate Road
Sandwich
Kent CT13 9NJ
UK

Wolfgang Huber
EMBL Outstation - Hinxton
European Bioinformatics Institute
Wellcome Trust Genome Campus
Hinxton
Cambridge, CB10 1SD
UK

Duncan Hull
University of Manchester
School of Computer Science
Oxford Road
Manchester M13 9PL
UK

Daniel Huson
University of Tübingen
Faculty of Computer Science
Chair of Algorithms in Bioinformatics
Sand 14
72076 Tübingen
Germany

Ravi Iyengar
Department of Pharmacology and Biological Chemistry
Mount Sinai School of Medicine
One Gustave L. Levy Place
New York, NY 10029
USA

Li Jin
Fudan University
Laboratory of Theoretical Systems Biology
School of Life Science
Handan Road 220
Shanghai 200433
China

Andreas Kämper
Max-Planck-Institute for Informatics
Computational Biology and Applied Algorithmics
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Teri E. Klein
Stanford University Medical Center
Department of Genetics
300 Pasteur Drive, Lane L301
Stanford, CA 94305-5120
USA

Oliver Kohlbacher
University Tübingen
Wilhelm Schickard Institute for Computer Science
Division for Simulation of Biological Systems
Sand 14
72076 Tübingen
Germany

Dennis Kostka
Max-Planck-Institute for Molecular Genetics
Ihnestrasse 63-73
14195 Berlin
Germany

Martin Krallinger
Protein Design Group (PDG)
National Biotechnology Center (CNB)
Campus Universidad Autónoma (UAM)
Cantoblanco 28049 (Madrid)
Spain

David C. Kulp
University of Massachusetts
Bioinformatics Research Laboratory
Computer Science Department
140 Governors Drive
Amherst, MA 01003
USA

ZoéLacroix
Arizona State University
Scientific Data Management Laboratory
P.O. Box 875706
Tempe, AZ 85287-5706
USA

Jerry Lanfear
Pfizer Global Research and Development
Pfizer Ltd
Ramsgate Road
Sandwich
Kent CT13 9NJ
UK

Thomas Lengauer
Max-Planck-Institute for Informatics
Computational Biology and Applied Algorithmics
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Hans-Peter Lenhof
Saarland University
Center for Bioinformatics
Building E11
P.O. Box 151150
66123 Saarbrücken
Germany

Azi Lipshtat
Department of Pharmacology and Biological Chemistry
Mount Sinai School of Medicine
One Gustave L. Levy Place
New York, NY 10029
USA

Phillip Lord
University of Manchester
School of Computer Science
Oxford Road
Manchester M13 9PL
UK

Claudio Lottaz
Max-Planck-Institute for Molecular Genetics
Ihnestrasse 63-73
14195 Berlin
Germany

Xing Jian Lou
Stanford University Medical Center
Department of Genetics
300 Pasteur Drive, Lane L301
Stanford, CA 94305-5120
USA

Bertram Ludäscher
University of California, Davis
Department of Computer Science
One Shields Avenue
Davis, CA 95616
USA

Avi Ma'ayan
Department of Pharmacology and Biological Chemistry
Mount Sinai School of Medicine
One Gustave L. Levy Place
New York, NY 10029
USA

François Major
Université de Montréal
Institute for Research in Immunology and Cancer
Computational and Theoretical Biology
2900, boulevard Édouard-Montpetit
Pavillon Marcelle-Coutu, Quai 20
Montreal QC H3T 1J4
Canada

Hans Matter
Sanofi-Aventis Deutschland GmbH
Chemical Science / Drug Design
65926 Frankfurt
Germany

Christian von Mering
University of Zurich
Institute of Molecular Biology
Bioinformatics Group
Winterthurerstrasse 190
8057 Zurich
Switzerland

William Stafford Noble
University of Washington
Department of Computer Science and Engineering
1705 NE Pacific Street
Seattle, WA 98195-7730
USA

John Overington
Inpharmatica Ltd
1 New Oxford Street
London WC1A 1NU
UK

Patricia M. Palagi
Swiss Institute of Bioinformatics
Proteome Informatics Group
CMU - 1, rue Michel Servet
1211 Geneva 4
Switzerland

Gaia Paolini
Pfizer Global Research and Development
Pfizer Ltd
Ramsgate Road
Sandwich
Kent CT13 9NJ
UK

Dariusz Przybylski
Columbia University
CUBIC, Department of Biochemistry
and Molecular Biophysics
1130 St. Nicholas Ave
New York, NY 10032
USA

John Quackenbush
Dana-Farber Cancer Institute
Department of Biostatistics and
Computationsl Biology
44 Binney Street, Sm822
Boston, MA 02115
USA

Jög Rahnenführer
Max-Planck-Institute for Informatics
Computational Biology and Applied
Algorithmics
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Matthias Rarey
University of Hamburg
Center for Bioinformatics Hamburg
(ZBH)
Bundesstrasse 43
20146 Hamburg
Germany

Udo Reichl
Otto-von-Guericke-University
Chair of Bioprocess Engineering
Universitätsplatz 2
39106 Magdeburg
Germany

Knut Reinert
Freie Universität Berlin
Institute for Computer Science
Computational Molecular Biology
Takustrasse 9
14195 Berlin
Germany

Ingo Reulecke
University of Hamburg
Center for Bioinformatics Hamburg
(ZBH)
Bundesstrasse 43
20146 Hamburg
Germany

Didier Rognan
Bioinformatics Group
Laboratoire de Pharmacochimie de
la Communication Cellulaire
CNRS UMR 7081
74, route du Rhin, B.P.24
67401 Illkirch
France

Kirsten Roomp
Max-Planck-Institute for Informatics
Computational Biology and Applied
Algorithmics
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Burkhard Rost
Columbia University
CUBIC, Department of Biochemistry
and Molecular Biophysics
1130 St. Nicholas Ave
New York, NY 10032
USA

Steven L. Salzberg
University of Maryland
Center for Bioinformatics and
Computational Biology
3125 Biomolecular Sciences Bldg
College Park, MD 20742
USA

Jean-Charles Sanchez
University of Geneva
Biomedical Proteomics Research
Group
Department of Structural Biology
and Bioinformatics
Centre Médical Universitaire
1, rue Michel Servet
1211 Genève 4
Switzerland

Heiko A. Schmidt
Center for Integrative Bioinformatics
Max F. Perutz Laboratories
Dr. Bohr Gasse 9
1030 Vienna
Austria

Stefan Schuster
Friedrich Schiller University
Department of Bioinformatics
Ernst-Abbe-Platz 2
07743 Jena
Germany

Yuri Sidorenko
Max-Planck-Institute for Dynamics
of Complex Technical Systems
Bioprocess Engineering
Sandtorstrasse 1
39106 Magdeburg
Germany

Ingolf Sommer
Max-Planck-Institute for Informatics
Computational Biology and Applied
Algorithmics
Stuhlsatzenhausweg 85
66123 Saarbrücken
Germany

Rainer Spang
Max-Planck-Institute for Molecular
Genetics
Ihnestrasse 63-73
14195 Berlin
Germany

Peter F. Stadler
University of Leipzig
Bioinformatics Group
Department of Computer Science
and Interdisciplinary Center for
Bioinformatics
Härtelstr. 16-18
04107 Leipzig
Germany

Robert Stevens
University of Manchester
School of Computer Science
Oxford Road
Manchester M13 9PL
UK

Paul Swift
Centre for Ecology and Hydrology,
Oxford
Molecular Evolution and
Bioinformatics Section
Mansfield Road
Oxford OX1 3SR
UK

Martin S. Taylor
University of Oxford
Wellcome Trust Centre for Human Genetics
Roosevelt Drive
Oxford OX3 7BN
UK

Philippe Thibault
Université de Montréal
Institute for Research in Immunology and Cancer
Computational and Theoretical Biology
2900, boulevard Édouard-Montpetit
Pavillon Marcelle-Coutu, Quai 20
Montreal QC H3T 1J4
Canada

Peter Uetz
The Institute for Genomic Research
9712 Medical Center Drive
Rockville, MD 20850
USA

Alfonso Valencia
Spanish National
Cancer Research Centre (CNIO)
Structural and Computational
Biology Program (S-CompBio)
Melchor Fernandez Almagro, 3
28029 Madrid
Spain

Martin Vingron
Max-Planck-Institute for Molecular Genetics
Ihnestrasse 63-73
14195 Berlin
Germany

Thomas Werner
Genomatix Software GmbH
Bayerstr. 85a
80335 München
Germany

Gareth Wilson
Centre for Ecology and Hydrology, Oxford
Molecular Evolution and
Bioinformatics Section
Mansfield Road
Oxford OX1 3SR
UK

Chris Wroe
University of Manchester
School of Computer Science
Oxford Road
Manchester M13 9PL
UK

Momiao Xiong
The University of Texas HSC at Houston
Human Genetics Center
School of Public Health
7000 Fannin, Suite 1200
Houston, TX 77030
USA

Michael Q. Zhang
Watson School of Biological Sciences
Cold Spring Harbor Laboratory
1 Bungtown Road
Cold Spring Harbor, NY 11724
USA

Part 1 Introduction

1
Bioinformatics – From Genomes to Therapies

Thomas Lengauer

1 Introduction

In order to set the stage for this book, this chapter provides an introduction to the molecular basis of disease. We then continue to discuss modern biological techniques with which we have recently been empowered to screen for molecular drugs targets as well as for the drugs themselves. The chapter finishes with an overview of the organization of the book.

2 The Molecular Basis of Disease

Diagnosing and curing diseases has always been and will continue to be an art. The reason is that man is a complex being with numerous facets, many of which we do not and probably will never understand. Diagnosing and curing diseases has many aspects, include biochemical, physiological, psychological, sociological and spiritual aspects.

Molecular medicine reduces this variety to the molecular aspect. Living organisms, in general, and humans, in particular, are regarded as complex networks of molecular interactions that fuel the processes of life. This "molecular circuitry" has intended modes of operation that correspond to healthy states of the organism and aberrant modes of operation that correspond to diseased states. The main goal of molecular medicine is the identification of the molecular basis of a disease, i.e. to answer the question: "What goes wrong in the molecular circuitry?". The goal of therapy is to guide the biochemical circuitry back to a healthy state. The molecular approach has already proven

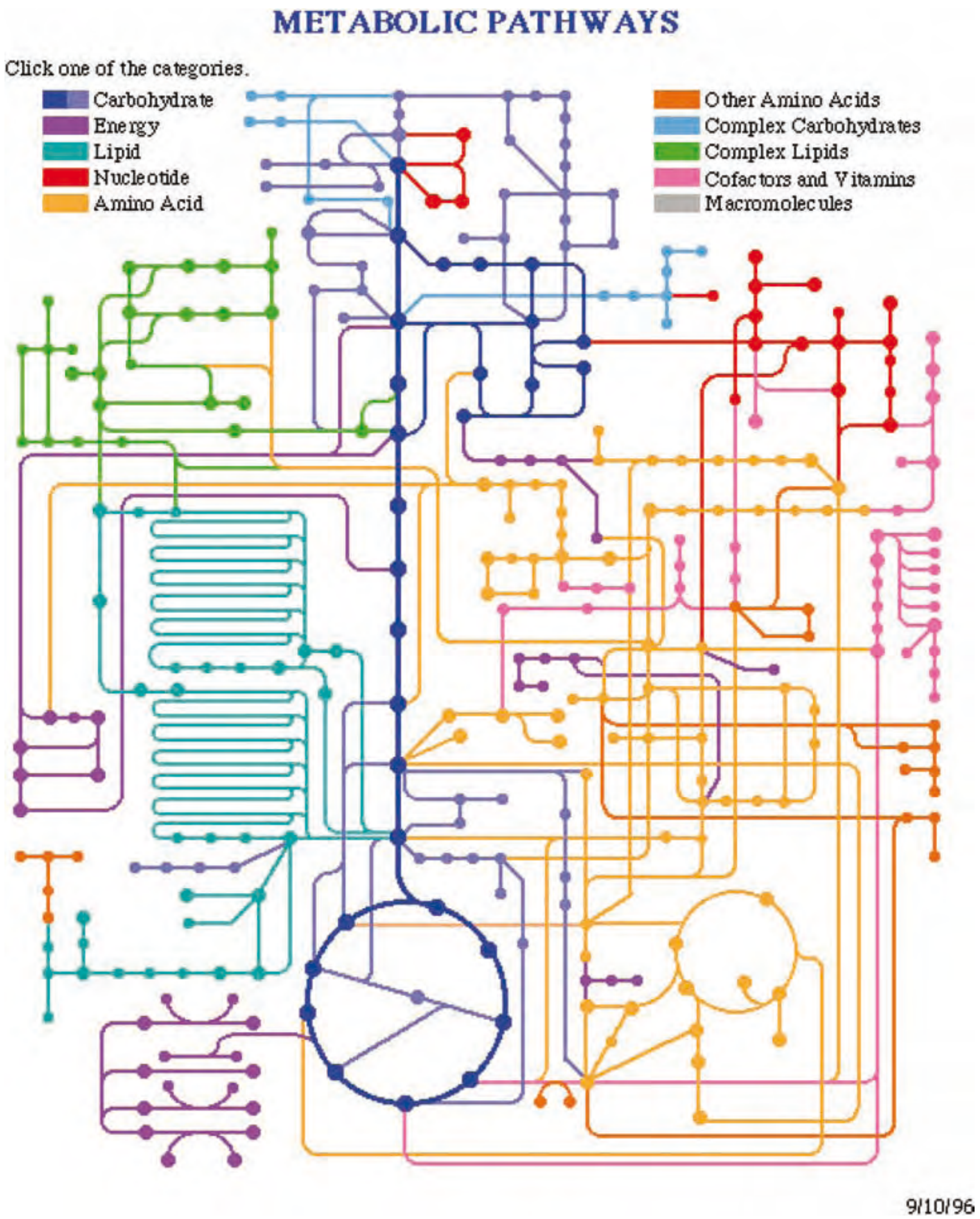
Bioinformatics – From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

its effectiveness for understanding diseases, and is dramatically enhanced by genomics and proteomics technology [5]. It is the prime purpose of this book to explore the contributions that this technology, particularly its computational aspect, can have to advancing molecular medicine.

As already noted, the molecular basis of life is composed of complex biochemical processes that constantly produce and recycle molecules, and do so in a highly coordinated and balanced fashion. The underlying basic principles are quite alike throughout all kingdoms of life, even though the processes are much more complex in highly developed animals and the human than in bacteria, for instance. Figure 1 gives an abstract view of such an underlying biochemical network, the *metabolic network* of a bacterial cell (the intestinal bacterium *Escherichia coli*) – it affords an incomplete and highly simplified account of the cell's metabolism, but it nicely visualizes the view of a living cell as a biochemical circuit. The figure has the mathematical structure of a *graph.* Each dot (*node*) stands of a small organic molecule that is metabolized within the cell. Alcohol, glucose and ATP are examples for such molecules. Each line (*edge*) indicates a chemical reaction. The two nodes connected by the edge represent the substrate and the product of the reaction. The colors represent the role that the respective reaction plays in metabolism. These roles include the construction of molecular components that are essential for life – nucleotides (red), amino acids (orange), carbohydrates (blue), lipids (light blue), etc. – or the breakdown of molecules that are not helpful or even harmful to the cell. Other tasks of chemical reactions in a metabolic network pertain to the storage and conversion of energy. (The blue cycle in the center of Figure 1 is the citric acid cycle.) A third class of reactions facilitates the exchange of information in the cell or between cells. This includes the control of when and in what way genes are expressed (*gene regulation*), as well as such tasks as the opening and closing of molecular channels on the cell surface, and the activation or deactivation of cell processes such as replication or apoptosis (programmed cell death). The reactions that regulate cellular processes are often collectively called the *regulatory network*. Recently, molecular networks that facilitate the propagation of signals within the cell are being selectively called *signal transduction networks*. Figure 1 only includes metabolic reactions, without any regulatory reactions or signal transduction cascades. Of course, all molecular networks of a cell are closely intertwined and many reactions can have metabolic as well as regulatory aspects. In general, much more is known about metabolic than regulatory networks, even though many relevant diseases involve regulatory rather than metabolic dysfunction.

The metabolic and regulatory networks can be considered as composed of partial networks that we call *pathways*. Pathways can fold in on themselves, in which case we call them *cycles*. A metabolic pathway is a group of reactions that turns a substrate into a product over several steps (pathway) or recycles



9/10/96

Figure 1 Abstract view of part of the metabolic network of the bacterium *E. coli* (from http://www.genome.ad.jp/kegg/kegg.html).

a molecule by reproducing it in several steps (cycle). *The glycolysis pathway* (the sequence of blue vertical lines in the center of Figure 1) is an example of a pathway that decomposes glucose into pyruvate. The *citric acid cycle* (the blue cycle directly below the glycolysis pathway in Figure 1) is an example of a cycle that produces ATP – the universal molecule for energy transport. Metabolic cycles are essential in order that the processes of life do not accumulate waste or deplete resources. (Nature is much better at recycling than man.)

There are several ways in which Figure 1 hides important details of the actual metabolic pathway. In order to discuss this issue, we have extracted a metabolic cycle from Figure 1 (see Figure 2). This cycle contributes to cell replication; more precisely, ;t is one of the motors that drive the synthesis of thymine – a molecular component of DNA. In Figure 2, the nodes of the metabolic cycle are labeled with the respective organic molecules and the edges point in the direction from the substrate of the reaction to the product. Metabolic reactions can take place spontaneously under physiological conditions (in aqueous solution, under room temperature and neutral pH). However, nature has equipped each reaction (each line in Figure 1) with a specific molecule that catalyzes that reaction. This molecule is called an *enzyme* and, most often, it is a protein. An enzyme has a tailor-made binding site for the transition state of the catalyzed chemical reaction. Thus, the enzyme speeds up the rate of that reaction tremendously, by rates of as much as 10^7. Furthermore, the rate of a reaction that is catalyzed by an enzyme can be regulated by controlling the effectiveness of the enzyme or the number of enzyme molecules that are available.

How does the enzyme do its formidable task? As an example, consider the reaction in Figure 2 that turns dihydrofolate (DHF) into tetrahydrofolate (THF). This reaction is catalyzed by an enzyme called *dihydrofolate reductase (DHFR).* The surface of this protein is depicted in Figure 3. One immediately recognizes a large and deep pocket that is colored blue (representing its negative charge). This pocket is a *binding pocket* or *binding site* of the enzyme, and it is ideally adapted in terms of geometry and chemistry so as to bind to the substrate molecule DHF and present it in a conformation that is conducive for the desired chemical reaction to take place. In this case, this pocket is also

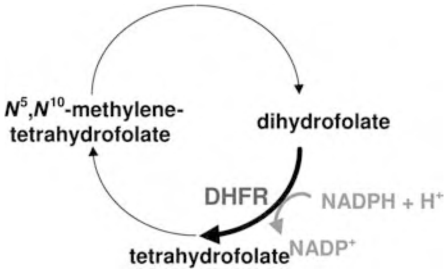


Figure 2 A specific metabolic cycle.

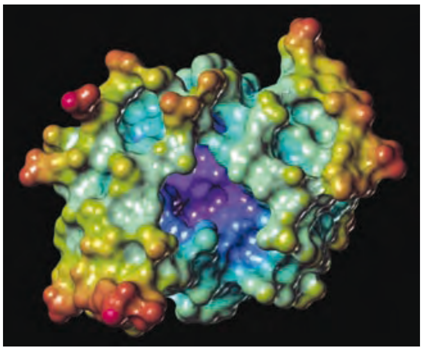


Figure 3 The 3-D structure of DHFR colored by its surface potential. Positive values are depicted in red, negative values in blue.

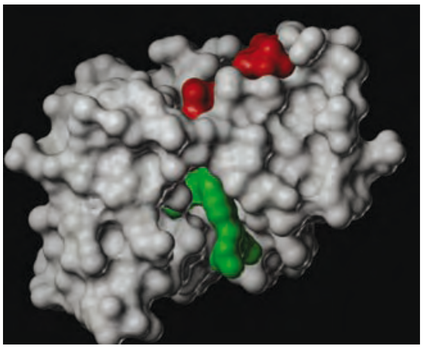


Figure 4 DHFR (gray) complexed with DHF (green) and NADPH (red).

where the reaction is catalyzed. We call this place the *active site*. (There can be other binding pockets in a protein that are far removed from the active site.)

There is another aspect of metabolic reactions that is not depicted in Figure 1 – many reactions involve *cofactors*. A cofactor is an organic molecule or a metal ion that has to be present in order for the reaction to take place. If the cofactor is itself modified during the reaction, we call it a *cosubstrate.* In the case of our example reaction, we need the cosubstrate NADPH for the reaction to happen. The reaction modifies DHF to THF and NADPH to $NADP^+$. Figure 4 shows the molecular complex of DHFR, DHF and NADPH before the reaction happens. After the reaction has been completed, both organic molecules dissociate from DHFR and the original state of the enzyme is recovered.

Now that we have discussed some of the details of metabolic reactions, let us move back to the global view of Figure 1. We have seen that each of the edges in Figure 1 represents a reaction that is catalyzed by a specific protein. (However, the same protein can catalyze several reactions.) In *E. coli* there are an estimated 1500 enzymes [6]; in human there are thought to be about least twice as many. The molecular basis of a disease lies in modifications of the action of these biochemical pathways. Some reactions do not happen at their intended rate (e.g. in gout), resources that are needed are not present in sufficient amounts (vitamin deficiencies) or waste products accumulate in the body (Alzheimer's disease). In general, imbalances induced in one part of the network spread to other parts. The aim of therapy is to replace the aberrant processes with those that restore a healthy state. The most desirable fashion in which this could be done would be to control the effectiveness of a whole set of enzymes in order to regain the metabolic balance. This set probably involves many, many proteins, as we can expect many proteins to be involved in manifesting the disease. Also, each of these proteins would have to be regulated in quite a specific manner. The effectiveness of some proteins would possibly have to be increased dramatically, whereas other proteins would have to be blocked entirely, etc. It is obvious that this kind of therapy involves a kind of global knowledge of the workings of the cell and a refined pharmaceutical technology that is far beyond what man can do today and for some time to come.

3 The Molecular Approach to Curing Diseases

For this reason, the approach of today's pharmaceutical research is far more simplistic. The aim is to regulate a single protein. In some cases we aim at completely blocking an enzyme. To this end, we can provide a drug molecule that effectively competes with the natural substrate of the enzyme. The drug molecule, the so-called *inhibitor*, has to be made up such that it binds more strongly to the protein than the substrate. Then, the binding pockets of most enzyme molecules will contain drug molecules and cannot catalyze the desired reaction in the substrate. In some cases, the drug molecule even binds very tightly (covalently) to the enzyme (suicide inhibitor). This bond persists for the remaining lifetime of the protein molecule. Eventually, the deactivated protein molecule is broken down by the cell and a new identical enzyme molecule takes its place. Aspirin is an example of a suicide inhibitor. The effect of the drug persists until the drug molecules themselves are removed from the cell by its metabolic processes and no new drug molecules are administered to replace them. Thus, one can control the effect of the drug by the time and dose it is administered.

There are several potent inhibitors of DHFR. One of them is *methotrexate (MTX)*. Figure 5 shows MTX (color) both unbound (left) and bound (right) to DHFR (gray). MTX has been administered as an effective cytostatic cancer drug for over two decades.

There are many other ways of influencing the activity of a protein by providing a drug that binds to it. Drugs interact with all kinds of proteins:

- With receptor molecules that are located in the cell membrane and fulfill regulatory or signal transduction tasks.

- With ion channels and transporter systems (again protein residing in the cell membrane) that monitor the flux of molecules into and out of the cell.

The mode of interaction between drug and protein does not always have the effect of blocking the protein. In some cases, the drug mimics a missing small molecule that is supposed to activate a protein. We call such drugs agonists.

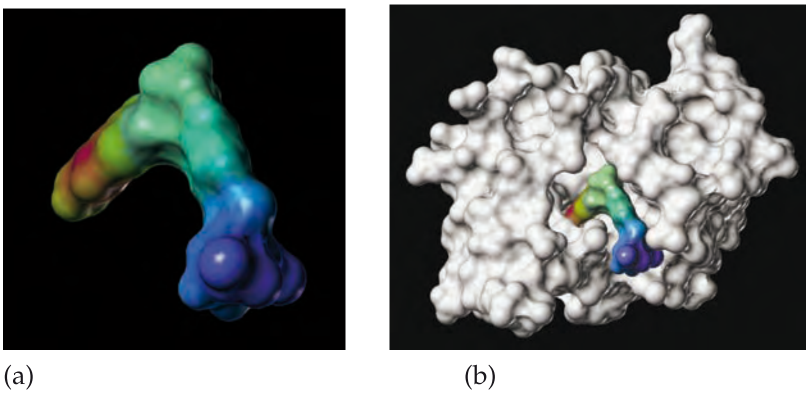


Figure 5 MTX (colored by its surface potential, see Figure 3): (a) unbound, (b) bound to DHFR (gray).

In general, we are looking for drugs that bind tightly to their protein target (effectiveness) and to no other proteins (selectivity).

Most drugs that are on the market today modify the enzymatic or regulatory action of a protein by strongly binding to it as described above. Among these drugs are long-standing, widespread and highly popular medications, and more modern drugs against diseases such as AIDS, depression or cancer. Even the lifestyle drugs that have come into use in recent years, e.g. Viagra and Xenical, belong to the class of protein inhibitors.

In this view, the quest for a molecular therapy of a disease decomposes into three parts:

- *Question 1: Which protein should we target?* As we have seen, there are many thousands of candidate proteins in the human. We are looking for one that, by binding the drug molecule, provides the most effective remedy of the disease. This protein is called the *target protein*.

- *Question 2: Which drug molecules should be used to bind to the target protein?* Here, the molecular variety is even larger. Large pharmaceutical companies have compound archives with millions of compounds at their disposal. Every new target protein raises the question of which of all of these compounds would be the best drug candidate. Nature uses billions of molecules. With the new technology of combinatorial chemistry, where compounds can be synthesized systematically from a limited set of b uilding blocks, this number of *potential* drug candidates is also becoming accessible to the laboratory.

- *Question 3: Given a choice of different drugs to administer to a patient, in order to alleviate or cure a specific disease, what is the best selection of drugs to give to that individual patient*? Questions 1 and 2 have been posed without the specifics of an individual patient in mind. Target protein and drug were selected for all putative patients collectively. With Question 3 we are entering the more advanced stage of *personalized medicine* – we want to understand the different ways in which different patients react to the same drug.

Question 3 has only come into the focus of research recently. The inclusion of the discussion of this question presents a major new feature of this book over its predecessor.

We will now give a short summary of the history of research on all three questions.

4 Finding Protein Targets

Let us start our discussion of the search for target proteins by continuing our molecular example of DHFR/MTX. As mentioned, DHFR catalyzes a reaction that is required for the production of thymine – a component of DNA. Thus, blocking DHFR impairs DNA synthesis and, therefore, cell division. This is the reason that MTX, an inhibitor of DHFR, is administered as a cytostatic drug against cancer. Is DHFR the "right" target protein in this context? The frank answer to this question must be "no". DHFR is active in every dividing cell, tumor cells as well as healthy cells. Therefore, MTX impairs the division of all dividing cells that it can get to. This is the cause of the serious side-effects of the drug such as loss of hair and intestinal lining. We see that in this case the limits of the therapy are mostly dictated by the choice of the wrong target protein. Why then is this protein chosen as a target? The answer to this question is also very simple: we cannot find a better one. This example shows how central the search for suitable target proteins is for developing effective drug therapies.

Target proteins could not really realistically be searched for until a few years ago. Historically, few target proteins were known at the time that the respective drug had been discovered. The reason is that new drugs were developed by modifying natural metabolites or known drugs, based on some intuitive notion of molecular similarity. Each modification was immediately tested in the laboratory either *in vitro* or *in vivo*. Thus, the effectiveness of the drug could be assessed without even considering the target protein. To this day, all drugs that are on the marketplace worldwide target an estimated set of not much more than 500 proteins [3]. Thus, the search for target proteins is definitely the dominant bottleneck of current pharmaceutical research.

Today, new experimental methods of molecular biology, the first versions of which were developed just a few years ago, provide us with a fundamentally new way – the first systematic way – of looking for protein targets. The basis for all of these methods was the technological progress made in the context of the quest for sequencing the human genome [1]. Based on this

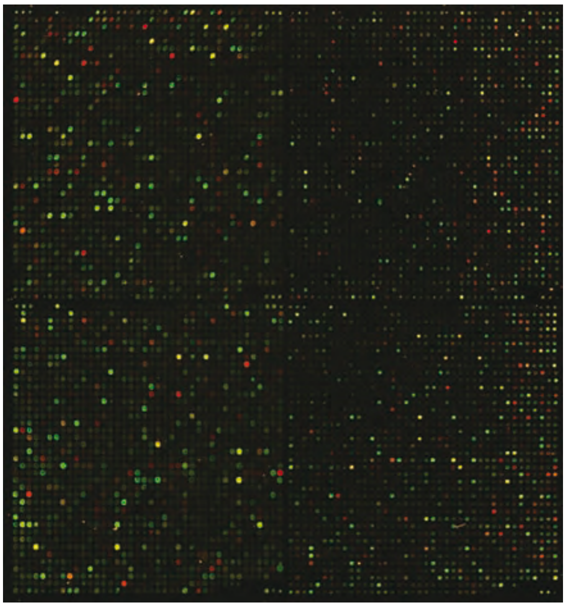


Figure 6 A DNA chip (from http://cmgm.stanford.edu/pbrown/explore/).

technology, additional developments have been undertaken to be able to measure the amount of expressed genes and proteins in cells. We exemplify this progress using a specific DNA chip technology [2]; however, the general picture extends to many other experimental methods under development.

Figure 6 shows a DNA chip that provides us with a differential census of the gene expressed by a yeast cell in two different cell states – one governed by the presence of glucose (green) and one by the absence of glucose (red). In effect the red picture is that of a starving yeast cell, whereas the green picture shows the "healthy" state. Each bright green dot indicates a protein that is manufactured (expressed) in high numbers in the "healthy" state. Each bright red dot indicates a protein that is expressed in high numbers in the starving cell. If the protein occurs frequently in both the healthy and the starving state, the corresponding dot is bright yellow (resulting from an additive mixture of the colors green and red). Dark dots indicate proteins that are not frequent, the tint of the color again signifies whether the protein occurs more often in the healthy cell (green), equally often in both cells (yellow) or more often in the diseased cell (red).

At this point, the exactly nature of the experimental procedures that generate the picture in Figure 6 is of secondary importance. What is important is how much information is attached to the colored dots in the picture. Here, we can make the following general statements.

(i) The identity of the protein is determined by the coordinates of the colored dot. We will assume, for simplicity, that dots at different locations also represent different proteins. (In reality, multiple dots that represent the same protein are introduced, on purpose, for the sake of calibration.) The exact arrangement of the dots is determined before the chip is manufactured. This involves identifying a number of proteins to be represented on the chip and laying them out on the chip surface. This layout is governed by boundary conditions and preferences of the experimental procedures, and is not important for the interpretation of the information

(ii) Only rudimentary information is attached to each dot. At best, the experiment reveals the complete sequence of the gene or protein. Sometimes, only short segments of the relevant sequence are available.

Given this general picture, the new technologies of molecular biology can be classified according to two criteria, as shown in the following two subsections.

4.1 Genomics versus Proteomics

In genomics, it is not the proteins themselves that are monitored, but rather we screen the expressed genes whose translation ultimately yields the respective proteins. In proteomics, the synthesized proteins themselves are monitored. The chip in Figure 6 is a DNA chip, i.e. it contains information on the expressed genes and, thus, only indirectly on the final protein products. The advantage of the genomics approach is that genes are more accessible experimentally and easier to handle than proteins. For this reason, genomics is ahead of proteomics, today. However, there also are disadvantages to genomics. First, the expression level of a gene need not be closely correlated with the concentration of the respective protein in the cell, although the latter figure may be more important to us if we want to elicit a causal connection between protein expression and disease processes. Even more important, proteins are modified post-translationally (i.e. after they are manufactured). These modifications involve glycosylation (attaching complex sugar molecules to the protein surface) and phosphorylation (attaching phosphates to the protein surface), for instance, and they lead to many versions of protein molecules with the same amino acid sequence. Genomics cannot monitor these modifications, which are essential for many diseases. Therefore, it can be expected that, as the experimental technology matures, proteomics will gain importance over genomics (see also Chapter 45).

4.2 Extent of Information Available on the Genes/Proteins

Technologies vary widely in this respect. The chip in Figure 6 is generated by a technology that identifies (parts of) the gene sequence. We are missing information on the structure and the function of the protein, its molecular interaction partners, and its location inside the metabolic or regulatory network of the organism. All of this information is missing for the majority of the genes on the chip.

There are many variations on the DNA chip theme. There are technologies based on so-called *expressed sequence tags (ESTs)* that tend to provide more inaccurate information on expression levels and various sorts of microarray techniques (see Chapter 24). All technologies have in common that the data they produce require careful quality control (Chapter 25). In general, it is simpler to distinguish different disease states from gene expression data (Chapter 26) than to learn about the function of the involved proteins from these data (Chapter 27). Proteomics uses different kinds of separation techniques, e.g. chromatography or electrophoresis combined with mass spectrometry, to analyze the separated molecular fractions (see Chapter 28). As is the case with genomics, proteomics technologies tend to generate information on the sequences of the involved proteins and on their molecular weight, and possibly information on post-translational modifications such as glycosylation and phosphorylation. Again, all higher-order information on protein function is missing. It is not feasible to generate this information exclusively in the wet laboratory – we need bioinformatics to make educated guesses here. Furthermore, basically all facets of bioinformatics that start with an assembled sequence can be of help. This includes the comparative analysis of genes and proteins (Chapter 37), protein structure prediction (Chapters 9–13), protein function prediction (Chapters 30–34), analysis and prediction of molecular interactions involving proteins (Chapters 16 and 17) as well as bioinformatics for analyzing metabolic and regulatory networks (Chapters 20–22). This is why all of bioinformatics is relevant for the purpose of this book.

If, with the help of bioinformatics, we can retrieve enough information on the molecular networks that are relevant for a disease, then we have a chance of composing a detailed picture of the disease process that can guide us to the identification of possible target proteins for the development of an effective drug. Note that the experimental technology described above is universally applicable. The chip in Figure 6 contains all genes of the (fully sequenced) organism *Saccharomyces cerevisiae* (yeast). The cell transition analyzed here is the diauxic shift – the change of metabolism upon removal of glucose. However, we could exchange this with almost any other cell condition of any tissue of any conceivable organism. The number of spots that can be put on a single chip goes into the hundreds of thousands. This is enough to put all of the human genes on a single chip. Also, we do not have to restrict ourselves to disease conditions; all kinds of environmental conditions (temperature, pH, chemical stress, drug treatment, diverse stimuli, etc.) or intrinsic conditions (presence or absence of certain genes, mutations, etc.) can be the subject of study.

The paradigm of searching for target proteins in genomics data has met with intense excitement from the pharmaceutical industry, which has invested heavily in this field over recent years. However, the first experiences have been sobering. It seems that we are further away from harvesting novel target proteins from genomics and proteomics data than we initially thought. However, in principle, a suitable novel target protein can afford a completely new approach to disease therapy and a potentially highly lucrative worldwide market share. For a critical review of the target-based drug development process, see Sams-Dodd [7].

Providing adequate bioinformatics support for finding new target proteins is a formidable challenge that is the focus of much of this book. However, once we have a target protein, our job is not done.

5 Developing Drugs

If the target protein has been selected, we are looking for a molecule that binds tightly to the relevant binding site of the protein. Nature often uses macromolecules, such as proteins or peptides, to inhibit other proteins. However, proteins do not make good drugs – they are easily broken down by the digestive system, they can elicit immune reactions and they cannot be stored for a long period of time. Thus, after an initial excursion into drug design based on proteins, pharmaceutical research has basically gone back to looking for small drug molecules. Here, one idea is to use a peptide as the template for an appropriate drug (peptidomimetics).

Due to the lack of fundamental knowledge of the biological processes involved, the search for drugs was, until recently, governed by chance. However, as long as chemists have thought in terms of chemical formulae, pharmaceutical research has attempted to optimize drug molecules based on chemical intuition and on the concept of molecular similarity. The basis for this approach is the lock-and-key principle formulated by Emil Fischer [4] over 100 years ago. Figures 3 to 5 illustrate that principle: in order to bind tightly, the two binding molecules have to be complementary to each other both sterically and chemically (colors in Figures 3 and 5). The drug molecule fits into the binding pocket of the protein like a key inside a lock. The lock-and-key principle has been the dominating paradigm in drug research ever since its proposal. It has been refined to include the phenomenon of induced fit, by which the binding pocket of the protein undergoes subtle steric changes in order to adapt to the geometry of the drug molecule.

For most of the past century the structure of protein-binding pockets has not been available to the medicinal chemist. Even to this day the structure of the target protein will not be known for many pharmaceutical projects for some time to come. For instance, many diseases involve target proteins that reside in the cell membrane and we cannot expect the three-dimensional (3-D) structure of such proteins to become known soon. If we have no information on the structure of the protein-binding site, drug design is based on the idea that molecules that are similar in terms of composition, shape and chemical features should bind to the target protein with comparable strength. The respective drug-screening procedures are based on comparing drug molecules, either intuitively or, more recently, systematically with the computer. The resulting search algorithms are very efficient and allow searching through compound libraries with millions of entries (Chapter 18).

As 3-D protein structures became available, the so-called *rational* or *structure-based* approach to drug development was invented, which exploited this information to develop effective drugs. Rational drug design is a highly interactive process with the computer originally mostly visualizing the protein structure and allowing queries on its chemical features. The medicinal chemist interactively modified drug molecules inside the binding pocket of the protein at the computer screen. As rational drug design began to involve more systematic optimization procedures interest arose in *molecular docking*, i.e. the prediction of the structure and binding affinity of the molecular complex involving a structurally resolved protein and its binding partner (Chapter 16). Synthesizing and testing a drug in the laboratory used to be comparatively expensive. Thus, it was of interest to have the computer suggest a small set of highly promising drug candidates. After an initial lead molecule has been found that binds tightly to the target protein, secondary drug properties have to be optimized that maximize the effectiveness of the drug and minimize side-effects (Chapter 19).

With the advent of *high-throughput screening* the binding affinity of as many as several hundred thousands drug candidates to the target protein can now be assayed within a day. Furthermore, *combinatorial chemistry* allows for the systematic synthesis of molecules that are composed of preselected molecular groups that are linked with preselected chemical reactions. The number of molecules that is accessible in such a combinatorial library can, in principle, exceed many billions. Thus, we need the computer to suggest promising sublibraries that promise to contain a large number of compounds that bind tightly to the protein (Chapters 16 and 18).

As in the case of target finding, the new experimental technologies in drug design require new computer methods for screening and interpreting the voluminous data assembled by the experiment. These methods are seldom considered part of bioinformatics, since the biological object, i.e. the target protein, is not the focus of the investigation. Rather, people speak of *cheminformatics* – the computer aspect of medicinal chemistry. Whatever the name, it is our conviction that both aspects of the process that guides us from the genome to the drug have to be considered together and we will do so in this book.

6 Optimizing Therapies

How is it that different patients react differently to the same drug? Reasons for this phenomenon can be manifold. Some are easier to investigate with methods of modern biology and bioinformatics than others. Here, we distinguish between infectious diseases and other diseases.

The molecular basis of any infectious disease is the interplay of a usually large population of a pathogen with the human host. The pathogen takes advantage of the human host or, in the case of virus, even hijacks the infected cells of the patient. Chapter 23 relates a story about the interplay of a viral pathogen with the infected host cell.

With infectious diseases, the drug often targets proteins of the infecting pathogen rather than the human host. The reason is the hope that drugs for such targets harbor less serious side-effects for the patient. However, in all infectious diseases, there is a constant battle going on between the host, whose immune system tries to eradicate the pathogen, and the pathogen that tries to evade the immune system. If the disease is treated with drugs, the administered drugs impose an additional selective pressure on the pathogen. On the road to resistance the pathogen constantly changes its genome and, thus, also the shape of the target proteins for drug therapy. Changes that are beneficial for the pathogen are those that render the drugs less effective, i.e. the pathogen becomes resistant. The results of this process are widely known. With bacteria, we observe increasingly resistant strains against antibiotic therapies (Chapter 41). With viral diseases such as AIDS the drug therapy has to be adapted continually to newly developing resistant strains within the patient (Chapter 40). Therapy selection must be individualized, in both cases, at least by taking the present strain of the pathogen into account and, at best, by also considering the individual characteristics of the host. Since the pathogen is a much simpler organism than the human host, the former is significantly easier than the latter.

Although the drug acts on its intended protein target, the drug has to find its way to the site of action and, eventually, has to be metabolized or excreted again. Along that path there are multiple ways in which the drug can interact with the patient. The resulting side-effects depend on the molecular and genetic status of the individual patient. Furthermore, the protein target often has different functions, such that its inhibition or agonistic activation can incur side-effects on molecular processes that were not intended to be changed. Again, the form and magnitude of such side-effects depends on the individual patient. This process of bringing about different reactions to drugs in different patient is much harder to analyze. The reason is that larger, often widespread, networks of interactions in the patient have to be taken in account. Analyzing them necessitates complex and accurately assembled patient histories and diverse molecular data that are seldom collected in today's clinical practice. Therefore, this approach to personalized medicine is still in its infancy (Chapter 39).

Another issue with diseases is the genetic predisposition of the human individual to the disease. Monogenetic diseases have been known for a long time and are relatively easy to analyze. Here, a defect in a single gene gives rise to the disease. However, these diseases are rare, in general. The major diseases like cancer, diabetes, and inflammatory and neurodegenerative diseases are based on a complex interplay between environmental and genetic factors with probably many genes involved. With data on the genomic differences between individuals just coming into being, the analysis of the genetic basis for complex diseases is embarking on a route that hopefully will lead to more effective means of prognosis, diagnosis and therapy.

7 Organization of the Book

This book is composed of three volumes. It is organized along the line from the genotype to the phenotype.

Volume 1: *The building blocks: sequences and structures*. This volume discusses the analysis of the basic building blocks of life, such as genes and proteins.

Volume 2: *Getting at the inner workings: molecular interactions*. This volume concentrates on the "switches" of the biochemical circuitry, the molecular interactions, as well as the circuits composed by these switches, the biochemical networks. In the former context, it partly also ventures into applied issues of drug design and optimization.

Volume 3: *The Holy Grail: molecular function*. This volume ties the elements provided by the first two volumes together and attempts to draw an integrated picture of molecular function – as far as we can do it today. The volume also discusses ramifications of this picture for the development and administration of drug therapies.

Each volume is subdivided in parts that are summarized below. The total book has 11 parts. *Volume 1* covers Parts 1–4.

Part 1 consists only of this chapter, and gives an introduction to the field and an overview of the book.

Part 2, consisting of Chapter 2, discusses bioinformatics support for assembling genome sequences. This is basic technology which is required to arrive at the genome sequence data that are the basis for much of what follows in the book. Major advances have been made in this area, especially during the finishing stages of completing the human genome sequence. The field has not lost its importance as we are embarking on sequencing many complex genomes, including over a dozen mammalian genomes. Furthermore, the technology is employed in projects that sequence closely related species, such as over a dozen species of *Drosophila*, in order to obtain a more effective database for functional genomics[1]. The authors of the chapter were part of the team that developed the assembler for the draft of the human genome sequence generated by Celera Genomics.

Part 3 is on molecular sequence analysis and comprises Chapters 3–8. Chapter 3 introduces the basic statistical and algorithmic technology for aligning molecular sequences. This technology forms the basis of much that is to follow. The author of the chapter has made seminal contributions to the field. Chapter 4 discusses methods for inferring ancestral histories from sequence data. This is one of the mainstays of comparative genomics. Similar to people, one can learn a lot about genes and proteins from looking at their ancestors and relatives, arguably more so with today's methods than from inspecting the gene or protein by itself. This attributes particular significance to this chapter in the context of this book. The authors of the chapter have made important contributions to the development of methods for inferring phylogenies and applied them to analyzing the evolution of *Homo sapiens*. Chapter 5 discusses the first major step from the genotype to the phenotype, i.e. the identification of protein-coding genes. The author of the chapter has developed one of the leading gene-finding programs. The ongoing debate on exactly what is the number of genes in the human chromosome years after the first draft of the human genome sequence was available shows that the issue of this chapter is still quite up-to-date. Furthermore, genes are a primary unit of linkage between the human genome and disease, as Chapter 38 discusses. Going into the gene's structure, most of the linkage with disease happens not in the coding regions of the genes that affect the structure of the coded protein. In general, proteins are far too well refined to be tampered with. Mostly, changing a protein means death to the individual

[1] see http://preview.flybase.net/docs/news/announcements/drosboard/GenomesWP2003.html for the respective community white paper

and only a few severe diseases (such as sickle cell anemia, Huntington's chorea or cystic fibrosis) are linked to changes in the coding regions of genes. More subtle influences of the genotype on disease involve polymorphisms in the noncoding regulatory regions of the disease gene that do not affect the structure of the protein, but the mechanism and level of its expression. This lends special importance to Chapter 6, which addresses bioinformatics methods for analyzing these regions. The author of the chapter has led the development of a widely used set of software tools for analyzing regulatory regions in genomes. The analysis of regulatory regions ventures into the more difficult to analyze noncoding regions of genes. However, the really dark turf of the human genome is presented by the long and mysterious repetitive sections. Up to 40% of the human genome is covered with these regions whose relevance (or irrelevance?) is under hot debate, especially since some of these regions seem to harbor potential silenced retroviral genes that may become active again at some suitable or unsuitable time. The identification of these regions (although not the elucidation of their function) is discussed in Chapter 7. The authors of this chapter have made seminal contributions and provided widely used software for computational gene finding, genome alignment and repeat finding. Chapter 8, finally, discussed the algorithmic and statistical basis of analyzing major genome reorganizations that happened as the kingdoms of life evolved, and that include splitting, fusing, mixing and reshuffling at a chromosomal level. Again, we are just beginning to understand the evolutionary role of these transactions. The author of this chapter has provided important contributions to the methodical and biological side of the field, many of them together with David Sankoff and Pavel Pevzner.

Part 4 of the book is on molecular structure prediction and comprises Chapters 9–15. The part starts with a chapter on a half-way approach to protein structure prediction which only aims at identifying the regions of secondary structure of the protein (α-helices and β-strands) and related variants. The resulting information on protein structure is very important in its own right and, in addition, helps guide or verify tertiary structure prediction. The authors of the chapter have made seminal contributions to protein structure prediction starting in the early 1990s that increased the prediction accuracy significantly (from around 65 to well over 70%).

The most promising approach to identifying the fold of a protein, today, selects a template protein from a database of structurally resolved proteins and models the structure of the protein under investigation (the target protein) after that of the template protein with sequence alignment methods. If the sequence similarity between the template and the target is high enough (roughly 40% or larger), then this alignment can even serve as a scaffold for providing a full-atom model of the protein structure. The respective structure prediction method is called homology-based modeling and is described in Chapter 10.

The author of this chapter has developed one of the most advanced homology-based structure prediction tools to date. If the sequence similarity between the template and target is below 40% then generating full-atom models for the target using the template structure becomes increasingly difficult and risky. In such low-sequence-similarity ranges aligning the backbone of the target protein to that of the template protein becomes the critical issue. If this is done correctly, one obtains a 3-D model of target backbone that can serve as an aid for structural classification of the target protein. Chapter 11 describes this process. The author of Chapter 11 has codeveloped a well-performing Internet server for this structural alignment task.

Homology-based modeling can only rediscover protein structures since it models the target on the basis of a known template structure. In *de novo* structure prediction, we try to come up with the structure of the protein, even if it is novel and has never been seen before. This subject is still a major challenge for the field of computational biology, but significant advances have been made in the past 10 years by David Baker's group (University of Washington, Seattle, WA) and the author of the chapter was one of the major contributors in this context. Today, there are several projects that aim at resolving protein structures globally, e.g. over whole proteomes. The approach is a combination of experimental structure resolution of a select set of proteins that promise to crystallize easily and fold into new structures, and homology-modeling other proteins using the thus increased template set. Chapter 13 describes these structural genomics projects. One author of the chapter codirects the Protein Data Bank (the main repository for publicly available proteins structures) and the other directs a major structural genomics initiative.

The last two chapters discuss structure prediction of another important macromolecule in biology – RNA. In contrast to DNA, which basically folds into a double-helical structure, RNA is structurally diverse. There is a well-understood notion of secondary structure in RNA, i.e. the scaffold that is formed by base pairs within the same RNA chain. This algorithmically and biologically well-developed field is presented in Chapter 14. The authors of the chapter have contributed a major software package for analyzing RNA secondary structures. The last chapter in this part looks at tertiary structure prediction for RNA, a comparatively much less mature field, and its author is one of the major experts in that field, worldwide.

Volume 2 covers Parts 5–7. Based on the knowledge about molecular building blocks afforded by Volume 1, Volume 2 ventures into questions of molecular function.

Part 5 starts by considering atomic events in molecular networks, i.e. the interactions between pair of molecules. Molecular interactions are important in two ways. First, understanding which molecules bind in an organism, when and how, is fundamental for understanding of the dynamic basis of life.

Second, as we have seen in the first parts of this chapter, modifying molecular interactions in the body with drugs is the main tool for pharmaceutical therapy of diseases. Drugs bind to target proteins. Understanding the interactions between a drug and its target protein is a prerequisite for rational and effective drug therapy. Part 5 addresses both these questions. The part comprises four chapters. Chapter 16 discusses protein–ligand docking, with the implicit understanding that the ligands of interest are mostly drugs or drug candidates. The chapter discusses how to computationally dock known ligands into structurally resolved protein-binding sites and also how to computationally assemble new ligands inside the binding site of a protein. The senior author is the developer of one of the most widely used protein–ligand docking tools, worldwide. Chapter 17 discusses molecular docking if both docking partners are proteins. This problem is of lower pharmaceutical relevance, as most drugs are small molecules and not proteins, but of high medical relevance, as the basis of a disease can often be an aberration of protein–protein binding events. Furthermore, the chapter also discusses protein–DNA docking, which is at the heart of gene regulation. (Here, the protein is a transcription factor binding to its site along the regulatory region of a gene, for instance.) The authors of this chapter have developed advanced software for protein–protein docking. The last two chapters in this part discuss problems in finding drugs. As described above, the drug design process is decomposed into a first step, in which a lead structure is sought, and a second step, in which the lead is optimized with respect to secondary drug properties. If the binding site of the target protein is resolved structurally, lead finding can be done by docking (Chapter 16). Otherwise, one takes a molecule that is known to bind to the binding site of the target protein as a reference and searches for similar molecules as drug candidates. Here, the notion of similarity must be defined suitably such that similar molecules have similar characteristics in binding to the target protein. Chapter 18 discusses this type of drug screening. Finally, Chapter 19 addresses the optimization of drug leads. The authors of Chapter 19 are from the pharmaceutical industry. They are experts in applying and advancing methods for drug optimization in the pharmaceutical context.

Part 5 has advanced considerably beyond fundamental research questions and into pharmaceutical practice.

In *Part 6* we take a step back towards fundamental research. This part discusses the biochemical circuitry that is composed of the kind of molecular interactions that were the subject of Part 5. Understanding these molecular networks is clearly the hallmark of understanding life's processes, in general, and diseases and their therapies, in particular. However, the understanding of molecular networks is in its infancy, and is not advanced enough, in general, to be directly applicable to pharmaceutical and medical practice. Still, the vision is to advance along this line and the four chapters in this part present various aspects of this process. Chapter 20 is on metabolic networks, the kind discussed in a little more detail in the beginning of this chapter. Metabolic networks are quite homogeneous with respect to the roles of the participating molecules. In general, we have a substrate that is converted to a product by a chemical reaction that is catalyzed by an enzyme, possibly with the aid of a cofactor. This homogeneity makes metabolic networks especially amenable to theoretical analysis. In addition, much is known about the topology (connection structure) of metabolic networks. However, we are still lacking much of the kinetic data needed to accurately simulate the dynamics of metabolic networks. The chapter presents methods for analyzing networks both statically and dynamically. The authors are among the main methodical contributors to the analysis of metabolic networks, worldwide. Chapter 21 analyzes gene regulation networks. These networks are more heterogeneous, since they incorporate different kinds of interactions – direct interactions, as when transcription factors bind to the regulatory regions of genes, and indirect interactions, as when transcription factors regulate the expression of genes that code for other transcription factors. Furthermore, proteins, as well as DNA and RNA, are involved in gene regulation. Inferring gene regulation networks necessitates much genomic information which is just on the verge of becoming available and, thus, the field is less mature than the area of analyzing metabolic networks. The author of Chapter 21 is one of the prime experts in the field of analyzing gene regulation networks. A very special type of molecular networks is concerned with transmission of information inside the cell. Usually, these signaling networks can be analyzed in terms of smaller modules than regulatory or metabolic networks. The special methods for analyzing these networks are presented in Chapter 22 by a group of outstanding experts in the field. Chapter 23 finally moves beyond the single cell and discusses interactions between a viral pathogen and its infected host cell – a major step from basic research to its application in a medical setting. This is a very young field and the author is one of its main proponents.

Part 7 is focused on a special types of experimental data that form the basis of much research (and debate) today – expression data. We have discussed the microarray (mRNA) expression data in the chapter above, when we addressed the quest of finding new target proteins for drug therapy. Expression data were the first chance to venture beyond the genome, which is the same in all cells of an organism, and analyze the differences between different cells, tissues and cell states. Therefore, these data have a special relevance for advancing molecular medicine and this justifies dedicating a separate part of the book with five chapters to them. Chapter 24 gives a summary of the whole field, from the experimental side of the technology of measuring mRNA expression and its implications on computational analysis methods to the bioinformatics methods themselves. Since expression data are typically quite noisy, with many sources of variance residing both in the technology (which can be improved, in principle) and the underlying biology (which can and should not be changed), issues of quality control of the data play a prominent role in this chapter. The author is a global expert in the field of analysis of expression data. The following four chapters go into more detail on computational issues. Chapter 25 presents statistical methods for pretreating the data so as to arrive at an optimally interpretable dataset and it is written by a leading group of researchers in the area. The following two chapters discuss two fundamentally different kinds of analysis of mRNA expression data. Chapter 26 discusses methods that analyze and group different datasets (microarrays), generated under different circumstances (e.g. from different patients or from the same patient at different time points). Such methods afford the distinction of healthy from sick individuals as well as the analysis of disease type and disease progression, thus providing effective help in disease diagnosis. Chapter 27 groups data differently. Here, we are not interested in distinguishing different experiments, but in understanding the roles of (groups of) genes in, say, the progression of a disease that has been monitored with a sequence of microarray experiments. The results of the analysis are supposed to afford insight into the disease process and clues for drug therapy. This is a much harder task than just grouping microarray datasets and it has turned out that it cannot be solved, in general, just on the basis of expression data. Therefore, this chapter also prepares for later chapters that discuss the analysis of gene and protein function in a more general context (Part 8). The authors of Chapters 26 and 27 participate in a joint German national project that aims both at advancing the methods, and at applying them to biological and medical datasets. mRNA expression data (so-called transcriptomics data, because the data assess the expression level of mRNA transcripts of genes) have the advantage of being generated comparatively easily, due to the homogeneous structure of DNA (to which the mRNA is backtranscribed before measuring expression levels). However, these data correlate only weakly with the expression level of the actual functional unit, i.e. the synthesized and post-translationally modified protein. Measuring expression directly at the protein level is a more direct approach, but experimentally significantly more challenging. Therefore, the state of the field of proteomics, which analyzes protein expression directly, is behind that of transcriptomics, as far the experimental side is concerned. Nevertheless, proteomics is rapidly emerging, with several promising experimental technologies and the respective computational methods for data assembly/analysis. Chapter 28 presents the state of this field. It is written by a leading academic group engaged in software development for the field of proteomics.

Volume 3 builds on Volumes 1 and 2, and aims at embarking along an integrated picture of molecular function, and its consequences for the development and administration of drug therapies. The volume covers Parts 8–11.

Part 8 comprises eight chapters and is devoted to molecular (mostly protein) function. We have has already addressed aspects of molecular function (e.g. the chapters on molecular interactions and molecular networks, as well as the chapters on expression data), and, along the way, it has become increasingly evident that molecular function is a colorful term that has many aspects and whose elucidation relies on many different kinds of experimental data. In fact, molecular function is such an elusive notion that we dedicate a special chapter to discussing exactly this term, and the way it is and should be coded in the computer, respectively. This Chapter 29 is written by two authors that are main proponents of advancing the state of ontologies for molecular biology. Then we dedicate four chapters to inferring information on protein function from different kinds of data: sequence data (Chapter 30), protein interaction data that are based on special experimental technologies that can measure whether proteins bind to each other or not, and do so proteome-wide, in the most advanced instances (Chapter 31), genomic context data, affording an analysis based on the comparison of genomes of many species (Chapter 32), and molecular structure data (Chapter 33). Since all of these data still do not cover protein function adequately, we add another chapter that addresses methods for inferring aspects of protein function directly from free text in the scientific literature (Chapter 34). Chapter 35 presents methods for fusing all the various kinds of information gathered by the methods presented in the preceding chapters to arrive at a balanced account of the available knowledge on the function of a given protein. Finally, Chapter 36 discusses the druggability of targets, i.e. the adequacy of proteins to serve as a target for drug design. This quality encompasses properties such as a suitable shape of the binding pocket to suit typical drug molecules and a certain uniqueness of the shape of the binding pocket, such that drugs that bind to this pocket avoid binding to other proteins that are not targets for the drug. Again, all of these chapters are written by outstanding proponents of the respective fields.

With Parts 1–8 we have covered the space from the genotype (the genome sequence) to the phenotype (the molecular function). However, we can still take additional steps to making all of this knowledge work in applied medical settings. This is the topic of *Part 9*. To this end, Part 9 focuses on the analysis of relationships and differences between genomes. In the first chapter, Chapter 37, the topic is rolled up in a general fashion by asking the question: "What can we learn from analyzing the differences between genomes?". Then, we focus on the medically most relevant differences between genomes of individuals of the same species. Specifically, we are interested in the human and in pathogens infecting the human. Chapter 38 discusses what we can learn from genetic differences between people about disease susceptibility. Chapter 39 then addresses the topic of personalized medicine: how can we learn from suitable molecular and clinical data how a patient reacts to a given drug treatment? The final two chapters address the evolution of pathogens in the human host (mostly to become resistant against the host's immune system and drug treatment). Chapter 40 discusses viral pathogens, specifically HIV, the virus that leads to AIDS. Chapter 41 covers the bacterial world. The authors of all chapters have made seminal contributions to the topic they are describing.

Part 10 is an accompanying section of the book that addresses important informatics technologies that drive the field of computational biology and bioinformatics. There are three chapters. Chapter 42 is on data handling. Chapter 43 discusses visualization of bioinformatics data; here, molecular structures are not the center of the discussion, since their visualization is in a quite mature state, but we focus on microscopic images data, molecular networks and statistical bioinformatics data. Chapter 44 focuses on acquiring the necessary computational power for performing the analysis from computer networks (intranets and the Internet). There is a special research community that provides the progress in the underlying informatics technologies and the authors of these chapters are outstanding proponents of this community.

In *Part 11*, finally, Chapter 45 addresses in a cursory manner emerging trends in the field that were too new at the time of the conceptualization of the book to receive full chapters, but have turned out to become relevant issue at the time that the book was written. Thus, this chapter gives a cautious and anecdotal look into the future of the field of bioinformatics.

The goal of this book is to provide an integrated and coherent account of the available and foreseeable computational support for the molecular analysis of diseases and their therapies. The authors that have contributed to the book represent the leading edge of research in the field. We hope that the book serves to further the understanding and application of bioinformatics methods in the fields of pharmaceutics and molecular medicine.

References

1 COLLINS, F. S., E. D. GREEN, A. E. GUTTMACHER AND M. S. GUYER. 2003. A vision for the future of genomics research. Nature **422**: 835–47.

2 DERISI, J. L., V. R. IYER AND P. O. BROWN. 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. Science **278**: 680–6.

3 DREWS, J. 2000. Drug discovery: a historical perspective. Science **287**: 1960–4.

4 FISCHER, E. 1894. Einfluss der Configuration auf die Wirkung der Enzyme. Ber. Dt. Chem. Ges. **27**: 2985–93.

5 PAPAVASSILIOU, A. G. Clinical practice in the new era. A fusion of molecular biology and classical medicine is transforming the way we look at and treat diseases. EMBO Rep. 2001. **2**: 80–2.

6 RILEY, M., T. ABE, M. B. ARNAUD, et al. 2006. *Escherichia coli* K-12: a cooperatively developed annotation snapshot – 2005. Nucleic Acids Res. **34**: 1–9.

7 SAMS-DODD, F. 2005. Target-based drug discovery: is something wrong? Drug Discov. Today **10**: 139–47.

Part 2 Sequencing Genomes

2
Bioinformatics Support for Genome-Sequencing Projects

Knut Reinert, Daniel Huson

1 Introduction

Even though the landscape of molecular data in biology has diversified significantly in the past decade, DNA sequence data still remain the principle basis of data collection and contribute to most bioinformatics analyses. Also, the field of bioinformatics was propelled to its current magnitude mostly by the rapid development in DNA-sequencing technology. Since experimental technology only allows the reading of short stretches of DNA, encompassing just a few hundred basepairs, the assembly of these pieces into contiguous chromosomes is still a major computational challenge.

In this chapter we first describe current assembly strategies for large genomes in Section 2. We then present some of the main algorithm problems and their treatment in Section 3 and give an overview of existing assemblers in Section 4.

2 Assembly Strategies for Large Genomes

2.1 Introduction

Humans have always been fascinated by the "secret of life", i.e. the question of how new organisms come into existence, how they develop from "nothing"? What is and where is the "blueprint", the set of instructions that determines the genesis of an animal or plant? In the course of the last century, science has begun to unravel parts of the puzzle. We know now that the instructions to build a complex organism are contained in each of its cells, encoded by a

Bioinformatics – From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

simple, yet fascinating mechanism. In 1928, Frederick Griffith, and, later, in 1944, Oswald Avery and coworkers pointed out that DNA (consisting of four very simple biochemical building blocks named adenine, cytosine, guanine and thymine) plays a vital role in heredity. In 1953, Francis Crick and James Watson discovered the double-helix structure of DNA which suggested that a simple linear sequence of nucleic acids gives rise to an intricate code for describing the blueprint of life. It was not until 1961 that researchers revealed the *genetic code* that employs codons, nonoverlapping triplets of nucleotides, to form a redundant code for the 20 amino acids that are the basic building blocks of proteins.

For a long time it was unthinkable to determine the actual sequence of nucleotides of a DNA molecule, i.e. to *sequence* a fragment of DNA. In the 1970s, a number of different approaches to sequencing DNA were pursued, and the method developed by Fred Sanger and his group prevailed. This method and other advances in biotechnology led to the sequencing of the 49kb-bacteriophage λ genome in 1982. For this work Fred Sanger was awarded the Nobel prize in chemistry in 1980, together with Walter Gilbert and Paul Berg. In the late 1980s, the question arose whether to attempt to determine the sequence of the human genome [42,48], a formidable technological challenge, given the huge size of the genome of approximately 3 billion base pairs. As a consequence, the Human Genome Project (HGP) [30] was established in 1990 to tackle the problem, armed with with a 15-year plan and a budget of approximately US$ 3 billion.

A major milestone in genome sequencing was achieved in 1995, when the 1.8-Mb genome of *Haemophilus influenza* was completed [14]. This was followed by the sequencing of other genomes, among them most notably that of yeast [35]. A main scientific issue in the 1990s was whether large eukaryotic genomes could be sequenced using a global "whole-genome shotgun assembly" (WGS) approach or whether such genomes needed to be broken down into smaller pieces and each piece sequenced separately. The assembly of the genome of the fruit fly (*Drosophila melanogaster*) in 2000 [38], of the human (*Homo sapiens*) in 2001 [54] and of the mouse (*Mus musculus*) in 2002 [37] demonstrated that the WGS approach is indeed feasible, and WGS has now become the leading paradigm.

The sequence of the human genome is of immense medical and biological importance. Significant advances in sequencing technology (in particular, the invention of the capillary gel sequencer), the availability of sufficient computational power and storage technology, and the existence of an appropriate algorithmic approach [56] inspired the founding of a private company, Celera Genomics, in 1998 with the stated goal of sequencing the human genome at a low cost and within a very short time.

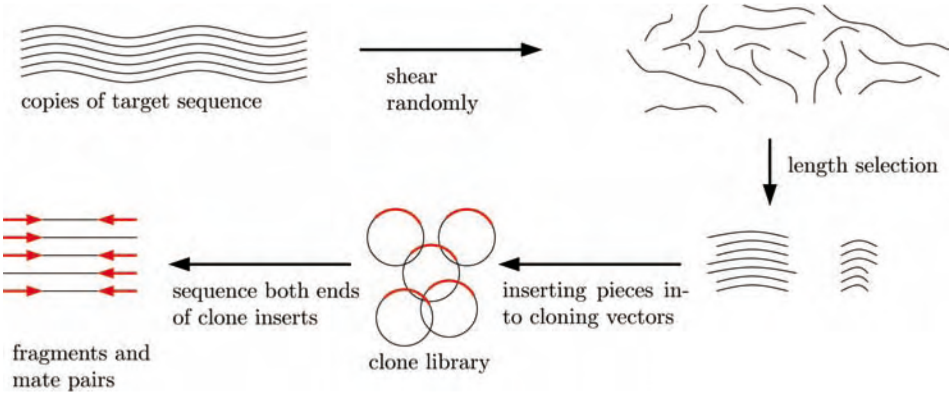

copies of target sequence
shear randomly
length selection
inserting pieces into cloning vectors
clone library
sequence both ends of clone inserts
fragments and mate pairs

Figure 1 Experimental protocol of paired-end shotgun sequencing.

This sparked off intense competition to produce a first assembly of the sequence of the human genome as quickly as possible, which led to the publication of two papers in February 2001 that both describe a draft sequence of the human genome [25,54].

All major sequencing projects are based on the same experimental technique, called *shotgun sequencing*. This technique is based on automated gel sequencers that use electrophoresis and fluorescent markers to determine the sequence of the nucleotides. The ability of these machines to read consecutive pieces of DNA degrades quickly with the length of the sequence and today a sequencing machine can read up to around 1000 consecutive base pairs of a fragment of DNA, depending on the degree of accuracy required. The sequence of a fragment determined in this way is called a *read*. The fragments are sampled from a stretch of DNA that is often referred to as the *source* sequence (the sequence that we take the fragment from) or as the *target* sequence (the sequence we want to reconstruct from the reads).

To determine the content of a long source sequence, one produces many copies of the source sequence (e.g. through cloning or growing colonies from a single progenitor) and then randomly breaks them into smaller pieces. Pieces of a given length are selected and one or both ends of such pieces are read by the automated sequencers. If both ends are read one does not only obtain the sequence at both ends of the piece of DNA, but also information about the relative orientation and distance of the two reads. This variant of shotgun sequencing was named *paired-end* shotgun sequencing by Myers and Weber [56], who also were the first to recognize the importance of collecting paired-end reads for sequence assembly. A pair of reads with associated distance information is called a *mate-pair*. Note that not all reads are in mate-pairs since the sequencing of one of the two mates can fail (see Figure 1 for an illustration

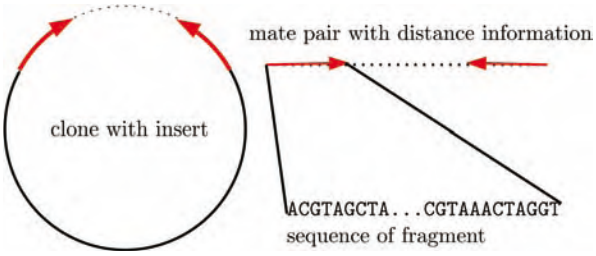


Figure 2 Fragments and mate-pairs.

of the shotgun sequencing process and Figure 2 for a more detailed illustration of a mate-pair).

The read and mate-pair information together with quality estimates of the data is fed into a computer program called an *assembler* that will attempt to reconstruct the original DNA source sequence. Note that there is no information on the location of any given read in the source sequence. However, by construction, many of the reads will come from overlapping regions of the source sequence and the first step in sequence assembly is to search for overlap alignments of high similarity between different reads. The pattern of overlaps between reads can be used to string together longer pieces of contiguous sequence, called *contigs*. The mate-pair information can then be used to order and orient sets of contigs with respect to each other, thus producing scaffolds. This process is called *sequence assembly*, and the resulting set of contigs and scaffolds is called an *assembly*. See Figure 3 for an illustration of the process.

Obviously, the large size of genomes makes sequence assembly a very difficult computational problem. Moreover, there are a number of additional difficulties. The read and mate-pair data contain errors and since DNA is a

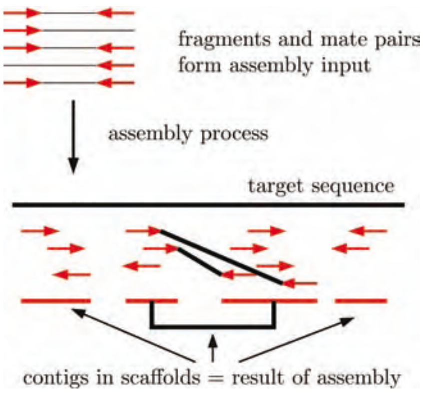


Figure 3 Pair-end reads are assembled into contigs based on how the reads overlap with each other. The contigs are then organized into scaffolds using the mate-pair information.

double-stranded molecule we do not know which strand a read is from. Also, a read may be chimeric, i.e. it may be the result of a fusion of two different pieces from different parts of the source sequence. Another problem is caused by polymorphism. If reads are taken from genomes of different individuals of the same species, they usually differ, in the case of humans at a rate of about 1 in 1000 bp. Even single organisms can be diploid or polyploid (i.e. they contain more than one copy of the same chromosome). Hence, if the data is acquired from different individuals that are not inbred, one must deal with a mixture of reads that come from seemingly different genomes. The largest difficulty is due to the fact that DNA sequences contain many different repeats of different size and fidelity. The detection and analysis of repeats is discussed in Chapter 7.

2.2 Properties of the Data

In this section we discuss some of the properties and error rates of the data generated in large genome-sequencing projects.

2.2.1 Reads, Mate-pairs and Quality Values

Sequencing large genomes is expensive, and over the past 10 years there has been a strong focus on developing faster, cheaper and more accurate ways of determining the sequence of DNA molecules. This includes substantial improvements in methods for DNA shearing, plaque and colony pickers, DNA template preparation systems, and, above of all, huge improvements in the throughput and data quality of automated sequencers (for a review, see Refs. [33,34]).

Most of the modern sequencers employ different fluorescent markers to distinguish between the four types of nucleotides. After a prefix of the fragment has left the sequencer, the marker attached to the last base of the prefix is excited by a laser and the resulting signal is measured. This analog measurement is converted into a digital base call. Each base determined in this way is assigned a quality value, given by $q = -10 \cdot \log_{10}(p)$, where p is the estimated error probability for the base [13]. For example, a quality value of 10 corresponds to an error rate of 1 in 10, whereas a quality value of 30 corresponds to an error rate of 1 in 1000. The value q is usually encoded in a single character that is stored together with the base character. Due to the nature of the sequencing process, it is clear that the distribution of the quality values is not uniform over the length of a read. The middle part usually has the best quality, whereas the quality drops at both ends of a read [12,13].

Older sequencers were slab-based and parallel sequencing lanes on an agarose gel were often mis-tracked, thereby generating incorrect mate-pairs. Modern capillary-based sequencers have eliminated this problem, but even with these machines human error (rotating or mislabeling of sequencing plates) can result in a wrong association of mate-pairs, leading to chimeric mate-pairs of unrelated reads. The error rate for mis-association of mate-pairs used to be about 10% for the older slab-based sequencers, but is now about 1%. Still, most assembly algorithms insist on the presence of more than one mate-pair to infer the relative ordering of two contigs.

In order to generate many copies of a fragment before sequencing, cloning vectors such as plasmids are used. The fragment is incorporated into the cloning vector and a sufficient number of copies is extracted after cloning.

A *spur read* is a read that aligns only partially to other reads from the same region of the source sequence. Spur reads can be the result of chimeric fragments that are obtained when two unrelated fragments fuse together during the creation of a clone library. They may also arise when fragments are contaminated with DNA from the linker or cloning vector.

To address these problems, the reads and mate-pairs obtained in the shotgun sequencing process are subjected to preprocessing steps that try to detect and remove most of the mentioned artifacts. For example, in a process called *vector and quality trimming* all reads are computationally inspected for pieces of the cloning vector genome and any traces of cloning vector sequence are removed. In addition, the quality values can be used to compute the expected number of errors in a window of the read. Any region (usually at the beginning or end of a read) for which this number is too high is then discarded. Such preprocessing steps will remove many of the artifacts, but not all. Hence, an assembly algorithm has to be able to cope with these problems to some degree.

2.2.2 **Physical Maps**

A physical map (see Ref. [47] for a description of the physical map used in the assembly of the human genome) of a genome G is given by the physical location of certain markers along G. The markers are used for navigation and can also be used for anchoring an assembly at its genomic coordinates. If parts of the target sequence are stored in clone libraries, then the correct order of the markers can be used to infer the order of the clones.

One can distinguish between two different families of methods for constructing a physical map [44]:

(i) *Restriction mapping.* Here one uses restriction enzymes to digest the DNA and then uses the lengths of the restriction fragments to reconstruct the positions of the restriction sites along the sequence. However, this technique works only for quite short genomic pieces.

(ii) *Fingerprint mapping.* Here we have a set of overlapping clones that we want to order based on common fingerprints. Therefore, one needs a set of clones that covers the target sequence redundantly. To determine which pairs of clones overlap with each other, we compute a fingerprint for each clone in such a way that overlapping clones have very similar fingerprints. The overlap information is used to order both the markers and the clones.

Fingerprints can be derived in a number of ways. One approach is to digest the DNA with a suitable restriction enzyme (e.g. *Hin*dIII was used in Ref. [25]) and use the restriction fragment sizes as a fingerprint. Alternatively, a whole restriction map of a clone can be used as a fingerprint.

Another way to obtain fingerprints is to use *STS* markers, which are short (200–500 bp) DNA sequences that occur exactly once in the given genome and are detectable by polymerase chain reaction (PCR). A number of other entities can also be used as markers. An EST is an *expressed sequence tag* that is derived from a cDNA [32]. It can be detected via a hybridization experiment or by PCR. The point is that one needs reliable, (essentially) unique markers, the presence of which is easily tested for. The assumption is that two clones overlap if they share a common set of markers.

Since the process of obtaining the fingerprints is error prone, it is very difficult to obtain a complete and accurate physical map of an entire genome. Physical maps are believed to have a high error rate of 10–20% [11] which makes the construction of a (correct) minimum tiling path a daunting task.

2.3 Assembly strategies

Given their higher complexity and larger size, it is not surprising that eukaryotic genomes are much more difficult to assemble than prokaryotic genomes. The assembly of a prokaryotic genome has become a routine task, whereas the assembly of a eukaryotic genome remains difficult. In the large sequencing projects of the last decade two different strategies were employed to determine the sequence of large eukaryotic genomes, i.e. the clone-by-clone (CBC) approach, which was used by the HGP to produce their assembly of the human genome [25], and the WGS approach, which was originally applied only to small genomes and was extended to large eukaryotic genomes by researchers at Celera Genomics [38, 54].

Both approaches are based on shotgun sequencing technology, but differ in an essential preparatory step. In the CBC approach, the target sequence is broken up into a redundant collection of overlapping pieces of an easily manageable size of approximately 100–150 kb. DNA molecules of this size can be incorporated into a vector such as a BAC and they are often referred to as *bacterial artificial chromosome (BAC)* clones. The problem of determining the sequence of a BAC clone is easily solved by using shotgun sequencing and subsequent assembly. For the human genome this step reduces the problem of assembling 3 Gb to approximately 40 000 small assembly problems, each of size around 100–150 kb. This is done with the help of available physical maps and computer programs [8,29].

The CBC approach has a number of advantages. Each individual assembly problem is easily solved, since the data sets are small and contain only local repeats. In a joint effort, work can be distributed by assigning different clones to different institutions for sequencing and assembly. The assembly itself is easier and can often be done without mate-pair information. However, during the course of the HGP it became evident that these advantages come at a high price. First, the physical maps used to place the location of assembled BAC clones are incomplete and have very high error rates. Second, since overlaps of the BACs are required to determine their order, there is a certain amount of redundant sequencing necessary, which results in higher costs. Third, one needs to construct many individual libraries of sequences for both the individual BAC clones and all their fragments. This allows the introduction of many artifacts; in particular, the creation of chimeric BAC clones. Fourth, it turned out that for the final assembly mate-pairs are necessary to improve the local ordering of contigs (see also description of current assemblers in Section 4). Finally, the assignment of sequencing a subset of the clones to different institutions using different protocols and standards leads to data of uneven quality.

The WGS approach is very bold. Rather than reducing the genome assembly problem to a large set of small BAC clone assembly problems, in this approach the shotgun strategy is applied to the whole genome. This method has essentially the opposite advantages and disadvantages of the CBC approach. The computational problem of assembling the reads is by no means trivial, requiring sophisticated algorithms, sufficient mate-pair information in the input and substantial computational resources. In particular, the assembler software has to cope with the full set of repetitive elements. However, the problem of mapping the resulting contigs and scaffolds to the genomic axis is not significantly more difficult than in the CBC approach. WGS data is much less effected by uneven sampling. The main advantage of this approach is that it is far easier to automate. Only very few libraries need to be created and sequenced, and all sequenced data is processed in a single computation, usually in an incremental fashion.

There was much debate over whether the WGS approach could possibly work for large eukaryotic genomes [17, 56]. However, the feasibility of the WGS approach in conjunction with paired-end reads as input was demonstrated by the assemblies of *D. melanogaster* [38], *H. sapiens* [54] and *M. musculus* [4, 26, 37]. WGS is now the predominant approach, and most current assembly programs are based on it (see Table 1 for an overview).

3 Algorithmic Problems and their Treatment

The sequence assembly problem is to reconstruct the sequence of a target DNA molecule from read and mate-pair information, in the presence of errors and repeats. The simplest mathematical formulation of this problem is the “shortest common super-string” (SCS) problem. Given a set of strings as input, the task is to find string s that contains all input strings as substrings and is shortest among all such super-strings.

Although this formulation is an extreme simplification of the sequencing problem, it is known to be NP-hard [15] and thus is believed to be impossible to solve optimally for large instances. A more sophisticated approach is to cast the problem as a maximum-likelihood problem [41], but this has not led to a deterministic approximation algorithm.

Current assemblers were developed using an engineering approach and are not designed to optimize some explicitly stated mathematical objective function. In this section we will discuss the fundamental tasks that any assembler program must address and we will outline some of the algorithmic approaches that are employed. The process of sequence assembly must address the following fundamental tasks:

(i) *Computation of overlaps in the presence of repeats.* To determine the layout of the reads on the genomic axis each assembly algorithm is based on the fact that the sequencing is redundant, in the sense that any given position in the sequence is covered by an average of x reads, where x is usually between 3 and 12. The value of x is called the *x-coverage*. Reads that were sampled from overlapping locations in the source sequence will exhibit a high scoring overlap alignment. The goal here is to determine which pairs of reads overlap. Unfortunately, reads may also exhibit a high-scoring overlap alignment if they stem from different instances of a repeat in the source sequence.

(ii) *Layout of reads.* Based on the overlap information, a second fundamental task is to determine a layout of the reads that overlap in a consistent way. This amounts to reconstructing nonrepetitive parts of the target sequence.

(iii) *Error correction and repeat resolution.* The goal here is to distinguish between sequencing errors and differences induced by the micro-heterogeneity of different instances of a repeat, and to attempt to reconstruct parts of the repetitive sequence.

(iv) *Layout of contigs using mate-pairs.* The goal here is to use mate-pair information to order and orient contigs relative to each other.

(v) *Computation the consensus sequence.* Finally, the sequence of each contig in the assembly must be determined.

In the following sections we will discuss the main methods for solving these tasks.

3.1 Overlap Comparison of all Reads

The input to an assembly program (or "assembler") is a set $F = \{f_1, \ldots, f_r\}$ of reads, together with mate-pair information and quality values. In order to assemble a set of reads, the assembler must be able to decide whether or not two reads f_i and f_j were sampled from overlapping locations in the source sequence.

Conceptually, this can be done by computing an overlap alignment between each pair of reads or their reverse complements. The detection of an overlap does not necessarily imply adjacency in the target sequence, since an overlap can be *repeat-induced*. In Figure 4 the regions marked R_1 and R_2 indicate two instances of the same repeat with near identical sequences. Hence, reads f_k and f_l form a repeat-induced overlap, whereas reads f_i and f_j form a *true* overlap.

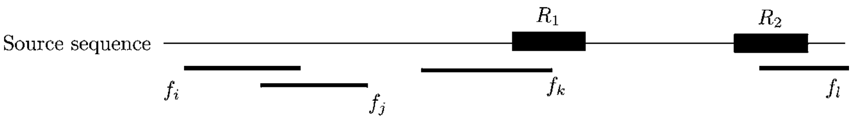


Figure 4 Reads that form true and repeat-induced overlaps. R_1 and R_2 indicate two instances of a repeat.

In a naive approach, one would require $O(r^2)$ sequence comparisons to determine all fragment overlaps. This is not feasible for large genomes where $r \approx$ 30–50 million. Since most reads do not actually overlap, this computational expense seems unnecessary. In fact, one can quickly reduce the number of required overlap computations to $O(r)$, by using the "seed-and-extend-and-refine" paradigm. All current assemblers use some version of this idea (Figure 5):

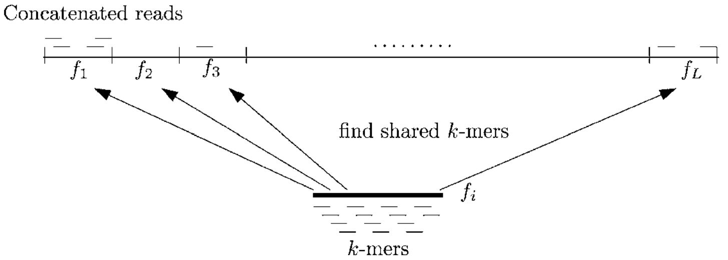


Figure 5 Overlap alignment of reads.

(i) Build a k-mer index H for all reads. This index maps any "k-mer" w (a word of length k) to the set of all occurrences of w in the reads. The value of k should be large enough such that in a random sequence of the same length as the target sequence, the expected number of k-mers is small. In contrast, k should not be too large so as to miss true overlaps due to sequencing errors. This index has two main applications:

   (a) If a pair of reads f_i and f_j do not share at least one k-mer (more sophisticated methods may have more complex requirements), they cannot possibly have a high fidelity overlap alignment and we need not attempt to compute one. If f_i and f_j contain one or more identical k-mers, these k-mers are referred to as *seeds* and the reads are candidates for an *extension*, which entails a more sophisticated and costly overlap computation (see Figure 5).

   (b) If a k-mer w appears significantly more often in the genome than expected, it probably lies in a repeat region of the genome. In this case, to avoid the computation of repeat-induced extensions, the k-mer is not used as a seed.

   A k-mer index can be computed in linear time and space. A first scan over all reads counts the number of k-mers. This allows us to efficiently allocate adjacent memory cells for all positions in the sequences that contain the same k-mer. In a second scan, the positions are written in the allocated positions (see, e.g. Refs. [5,45]).

(ii) The second phase is an extension phase, which makes use of the k-mers computed in the seed phase. Most ideas used here are very similar to BLAST [1,40]. Usually one combines two or more k-mer hits that are near to each other. Then the local alignment is extended in both directions until the quality of the extension starts to deteriorate. The result of this phase is a set of local alignments (they are depicted as longer black diagonals in Figure 6).

(iii) Finally, most algorithms end this stage by refining a set of local alignments with a fast version of the global Needleman–Wunsch algorithm. This can be done by using the shared k-mer information in a number of ways. (i) One can obtain a bound on the quality of the alignment and use it to compute a banded alignment [7]. (ii) One can compute an alignment allowing only k mismatches [6, 39, 40]. (iii) One can use the position of the shared k-mers and compute a chain of the local alignments from the extension phase together with smaller local alignments between their ends (see Figure 6 for an illustration).

The described "overlap" phase of assembly produces a collection of pairwise overlaps between reads which predominantly consists of true positive

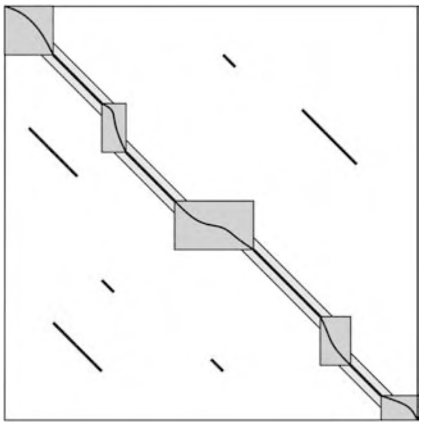


Figure 6 Seed, extend and refine paradigm. First, k-mer seeds being extended, then a banded alignment is computed that explores narrow bands around the extended seeds and, possibly, larger regions between them.

overlaps (Figure 7), i.e. overlaps that result from the fact that the involved reads stem from overlapping positions in the target sequence. However, there will be a number of false positives (repeat-induced overlaps) and false negatives (which may result from the seed-and-extend strategy missing an overlap due to sequencing errors). In Section 3.3 we will discuss how true and repeat-induced overlaps can be used to correct sequencing errors and to classify different repeat instances.

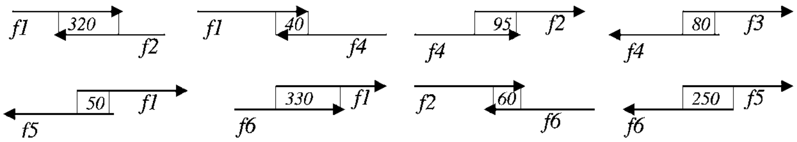


Figure 7 A collection of pairwise overlapping reads

We can view the collection of overlaps in terms of an edge-weighted, semi-directed graph, the *overlap graph* $OG(\mathcal{F})$ (Figure 8). There are two types of edges in this graph. A directed *read-edge* represents a read; the source and target nodes of the edge corresponding to the $5'$ and $3'$ ends of the read, respectively. The *weight* of a read-edge is simply the length of the corresponding read.

An *overlap-edge* represents an overlap between two reads and joins the two appropriate vertices contained in the corresponding read-edges. The weight of an overlap-edge is set to the negative length of the overlap. If the overlap corresponds to a gapped alignment of the ends of two reads, the amount of

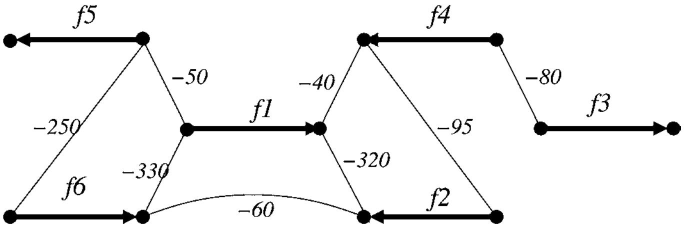


Figure 8 Overlap graph corresponding to the collection of overlaps in Figure 7. Read-edges are shown in bold.

overlap can be more accurately represented by a pair of numbers indicating the length of the two subsequences involved in the alignment.

3.2 Contig Phase: Layout of Reads

Ideally, in the absence of repeat-induced overlaps each connected component C of the overlap graph $OG(\mathcal{F})$ will correspond to a collection of reads sampled from the same local region of the target sequence. However, in practice, due to the abundance of long-range repeats, the overlap graph always consists of one huge, highly connected component.

The goal of the layout phase is to determine sets of reads that possess a consistent layout. Here, a *layout* is defined as an assignment of coordinates to all nodes of C that specify the start position s_i and the end position e_i of each read f_i in C. A layout is called *consistent* if every overlap-edge e is realized in the layout, which is the case when the coordinates assigned by the layout induce the corresponding overlap of the two appropriate reads. A layout is called *correct* if the relative positioning of the reads in the layout corresponds to their relative positioning in the source sequence. Any layout represents the reconstruction of a stretch of contiguous sequence in the target genome (a contig).

A read f_i is said to be *contained* in another read f_j if f_i is equal or highly similar to an internal portion of f_j or the reverse complement of f_j. Since contained reads contribute no additional overlap information, they are usually set aside in the layout phase of assembly. They are brought back into play later to contribute to the computation of arrival statistics and to the scaffolding of contigs using mate-pairs.

The problem of determining a minimal consistent layout of a set of overlapping reads is equivalent to the SCS. As the latter problem is known to be NP-hard [15], assemblers use heuristics to address the problem.

One widely used heuristics greedily "selects" a subset of overlap-edges S such that the union of S and the set of all read-edges F defines an alternating path of reads and overlaps that spans the set of read-edges. Initially, the edge

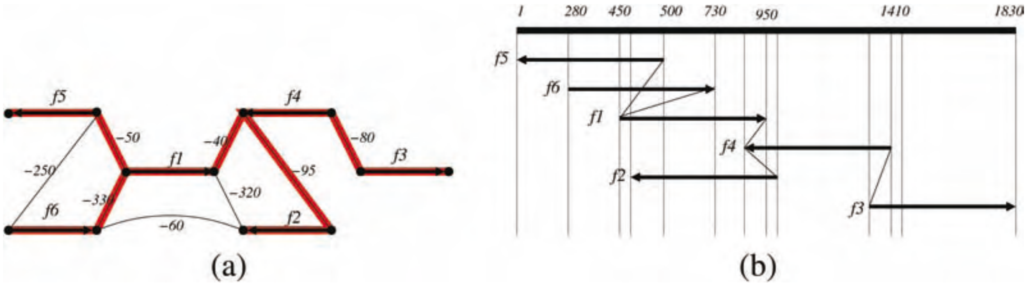


Figure 9 (a) An example of an overlap graph for six reads $\{f_1, \ldots, f_6\}$ that are as assumed to overlap as indicated in Figure 7. The edges of a maximal spanning tree are highlighted. (b) The layout of the reads is defined by the maximal spanning tree.

representing the longest overlap is selected. Then, all overlap-edges in the overlap graph are considered in ascending order of length of overlap. An overlap-edge e is selected if neither of the two nodes of e is already incident to a selected overlap edge.

Another simple heuristics for assigning the coordinates to a component C is to compute a *maximal spanning tree* that includes all read-edges, and maximizes the amount of overlap between reads (Figure 9).

In the presence of repeat-induced overlaps, any read that spans a repeat boundary may potentially overlap with reads from unrelated regions of the genome and thus bring them together in the same component C of the overlap graph. In this case, some of the overlap-edges in C will represent true overlaps, while others will represent repeat-induced overlaps. Both heuristics described above will fail to produce a correct layout whenever they utilize one or more repeat-induced overlap-edges.

As discussed before, many repeat-induced overlap-edges can be avoided in the overlap phase. To alleviate the problem further, one can attempt to distinguish between true overlaps and repeat-induced overlaps by taking a closer look at the overlap alignment. A number of mismatches in the alignment that is significantly higher than expected for the given level of sequencing error indicates that the two reads come from different instances of an inexact repeat, ideally taking the quality values into account.

Once a layout has been computed, a closer study of a multiple alignment of the reads in the layout may yield additional information, provided that sequencing errors will be randomly distributed, whereas repeat-induced discrepancies will occur in a correlated fashion. This is discussed in more detail below.

The most useful combinatorial insight is that if the reads contained in a connected component C of the overlap graph were recruited from different instances of a repeat and if some of the reads span the repeat boundaries, then

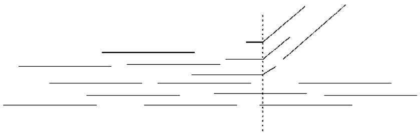


Figure 10 From left to right the reads overlap consistently until we reach the "branch point" at the position indicated by a dotted line. From this position onward, the data is partitioned into two incompatible chains of overlapping reads. Here, the reads on the left of the branch point lie in the interior of a repeat, whereas the reads that span the branch point overlap with a unique flanking sequence.

the latter reads will give rise to inconsistencies in the layout. That means, there will be overlap-edges in C that are not compatible with the overlaps induced by the layout. These incompatible overlaps will involve those reads that span repeat boundaries and a detailed analysis of the pattern of overlaps will uncover potential branch points in the layout (Figure 10).

A *branch point* is a position of a layout within a read at which a single consistent chain of overlapping reads possesses at least two different and mutually exclusive extensions. Whenever a branch point is detected, the adjacent overlaps are removed from the graph and, consequently, the connected component C is partitioned into smaller components, each giving rise to an individual contig.

As mentioned above, a consistent layout of reads defines a contig, which in this case is also called a *unitig* ("*uni*quelly assemble-able con*tig*"), as any given set of reads possesses at most one consistent layout.

Ideally, any unitig u computed in the layout phase will represent a unique stretch of the source sequence and will consist only of reads from that region. We refer to a unitig of this type as a *unique*-unitig or *U-unitig* (Figure 11). Alternatively, and in the absence of inconsistent overlaps, a unitig u may also represent a stretch of sequence that is repeated twice or more in the source sequence and may consist of reads collected from different instances of the repeat.

Methods for distinguishing between U-unitigs and non-unique unitigs make use of the sequencing coverage. For a given level of sequencing coverage, we can work out how many reads we expect to see in a unitig of a given length under the assumption that the unitig represents a unique stretch of the source sequence or that the unitig represents repetitive sequence, respectively.

In other words, a non-unique unitig can often be detected because it contains significantly more reads than expected. Let r be the number of reads and G be the estimated length of the source sequence. It can be shown [31] that for a unitig consisting of r reads and of approximate length ρ, the probability of

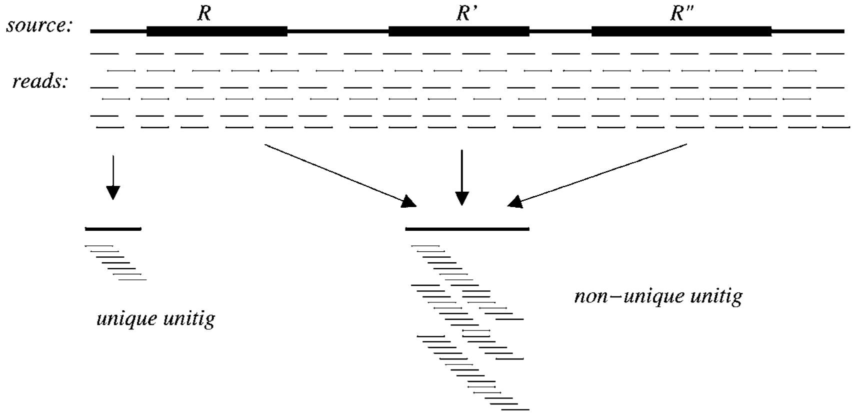


Figure 11 A unitig represents a chain of consistently overlapping reads. However, a unitig does not necessarily represent a segment of unique source sequence. For example, its fragments may come from the interior of the different instances of a long repeat, as shown here. R, R' and R'' represent three instances of the same repeat.

seeing $k-1$ start positions in an interval of length ρ is:

$$\frac{e^{-c}c^k}{k!},$$

with $c := \frac{\rho r}{G}$, if the unitig is not oversampled, and:

$$\frac{e^{-2c}(2c)^k}{k!},$$

if the unitig consists of reads recruited from two instances of a repeat. The *arrival statistic* is the log of the the ratio of these two probabilities:

$$c - (\log 2)k.$$

In practice, a unitig is considered to be unique if its arrival statistic is 10 or above, say.

3.3 Error Correction and Resolving Repeats

In the previous section we discussed how a layout of reads can be collapsed into a contig and how one can detect inconsistencies in the layout that indicate repeat boundaries or how arrival statistics can be used to classify contigs as repetitive.

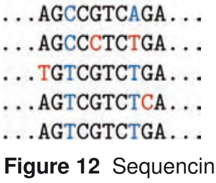
In this section, we use similar techniques, but with a different goal. Branch-point detection only determines the boundary of a repetitive region to a unique region in the genome and an arrival statistic can merely point to problematic regions. Error correction and repeat resolution approaches take a closer look at the distributions of errors in the layout of a collection of reads and their main task is to determine whether a mismatch in a pairwise alignment is due to a sequencing error, a single nucleotide polymorphism (SNP) or a low copy repeat.

The errors in a repetitive contig and the errors in a nonrepetitive contig are differently distributed. In a nonrepetitive contig errors in overlaps can be explained by sequencing errors which should occur independently from each other in each read. In contrast to this, repetitive contigs by definition consist of reads that are from instances of a repeat from different genomic locations. Depending on the nature of a repeat, two instances differ from each other by a certain amount.

In order to be able to classify sequences as repetitive or nonrepetitive, one needs a suitable null model, i.e. the sequencing error rate in the local genomic region. This error rate was often assumed to be a constant that could be refined using bootstrapping methods [9]. Alternatively, it was estimated using the quality values of bases in the reads. Huang [22] estimated the amount of sequencing errors in a local neighborhood based on the overlaps of an individual read with its overlapping partners (see also Ref. [21]). Developing this idea further, one could obtain an even better estimate of the error rate by constructing a multiple alignment in the layout phase. Such approaches work well if no additional source of error confuses the estimation of the sequencing error. If, however, repetitive overlaps are present, then these approaches cannot be applied directly. Nevertheless, we can assume that we have a rather good idea of the sequencing error for a collection of overlaps.

The fact that the reads are collapsed into a contig means that this difference is small, i.e. in the range of 1–3%. This is still significantly higher than the assumed rate of SNPs and hence this *microheterogeneity* can be used for detecting the different repeat instance (Figure 12).

This simply means that we use the fact that an instance of a repeat differs slightly from other instances. Hence, reads from a certain genomic location *always* differ from the reads in the other location, except in the unlikely event that the corresponding positions are changed by a sequencing error. Some



```
...AGCCGTCAGA...
...AGCCGTCAGA...
...AGCCCTCTGA...
...TGTCGTCTGA...
...AGTCGTCTCA...
...AGTCGTCTGA...
```

Figure 12 Sequencing errors (in red) and micro-heterogeneity of a collapsed repeat (in blue).

assembly programs like Euler [43] and ARACHNE [4] have a built-in, simple error correction phase that corrects numerous mistakes.

However, since the problem is modular, several papers addressed it individually. In Figure 12 differences caused by repeats are shown in blue and differences caused by sequencing errors are shown in red. The blue columns are called *DNPs* (defined nucleotide positions) [52] or *separating columns* [28] and can be used to separate the individual copies of a repeat.

The method of Tammi and coworkers proceeds in a straightforward way. It first prepares multiple alignments which it then refines, using a realignment algorithm [2]. Then, the consensus base in a column is defined as the most frequent base of the column. Whenever we see a certain number of *coinciding* differences from the consensus, the column is a candidate for usage in repeat separation (e.g. the first blue columns in Figure 12). If another candidate column can be found, these candidate columns define a DNP (e.g. the second blue column Figure 12).

Since the above procedure identifies errors that are due to micro-heterogeneity in repeats, we can attribute the remaining errors to the sequencing phase. Hence, the DNPs can also be used for correcting sequencing errors [51].

3.4 Layout of Contigs

In the layout phase, reads are assembled into contigs based on their overlaps, as reported in the overlap graph. Ideally, one may hope that this will give rise to a small number of very large contigs, perhaps one per chromosome arm. However, due to two problems this cannot happen. (i) Shotgun sequencing produces a random sampling of the source sequence, thus the coverage fluctuates along the sequence and some regions will remain unsampled, giving rise to *sequencing gaps* (see Ref. [31] for a mathematical treatment of the statistics for the length and number of such gaps). (ii) Repeats in the source sequence lead to the break-up of potential contigs into smaller ones, as described above.

Hence, a common strategy is to arrange sets of contigs into so-called *scaffolds* or *super-contigs* with the help of mate-pair information. More precisely, a scaffold consists of an ordered list of contigs $(c_1, c_2, \ldots, c_t)$, alongside a specification of the orientation of each individual contig (i.e. whether to use c_i or the reverse complement $\bar{c}_i$) and an estimation of the distance between any two consecutive contigs. A scaffold is deemed *correct* if the relative positioning and orientation of its contigs corresponds to the true locations in the source sequence.

As described above, shotgun sequencing projects often use a *paired-end* or *double-barreled* shotgun protocol, in which clones of a given fixed length are sequenced from both ends. This approach produces pairs of reads, called

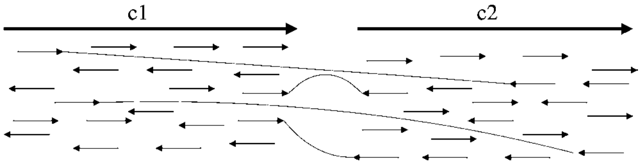


Figure 13 If two assembled contigs c_1 and c_2 correspond to neighboring regions of the source sequence, then we can expect to find mate-pairs that span the gap between them.

mate-pairs, whose relative orientation and mean distance l (with standard deviation σ) are known (Figure 2).

Standard size-selection techniques are used to produce a collection (library), of clones that have approximately the same length. A typical mixture of clone lengths is $l = 2$, 10 and 150 kb. With care, a standard deviation σ of approximately $1/10$ of l can be achieved.

Consider two contigs c_1 and c_2 produced in the layout phase of assembly. If they correspond to neighboring regions in the source sequence, we can expect to find mate-pairs that span the gap between them, as indicated in Figure 13. Such mate-pairs can be used to determine the relative orientation and estimated distance between c_1 and c_2.

Assume that the two contigs c_1 and c_2 are connected by mate-pairs $m_1, m_2, \ldots, m_k$. Each mate-pair provides an estimate of the distance between the two contigs. If these estimates are viewed as independent measurements, then they can be combined into a single estimate using standard statistical calculations.

As the assignment of reads to their mates is error prone, the existence of a single mate-pair linking two different contigs is not deemed significant. It is, however, of great statistical significance if two U-unitigs c_1 and c_2 are linked by two different mate-pairs in a consistent way. Similarly it is very unlikely that two mate-pair specification errors would put together two pairs of reads from the same two local regions of the source genome.

Assume that we are now given a collection of contigs $\{c_1, c_2, \ldots, c_k\}$ and a table of mate-pair information that links pairs of reads that are embedded in the contigs. To discuss the problem in more detail, we introduce the *contig-mate* graph. In this graph, each contig c_i is represented by a directed *contig-edge* having nodes s_i (the start node) and e_i (the end node). So-called *mate-edges* are added between such nodes to indicate that the corresponding contigs contain reads that are mates. For example, the two contigs c_1 and c_2, together with the collection of mates depicted in Figure 14 give rise to the contig-mate graph indicated in Figure 15.

If a set of different mate-pairs link two different contigs c_1 and c_2 in a consistent manner, then the contig-mate graph can be simplified by replacing

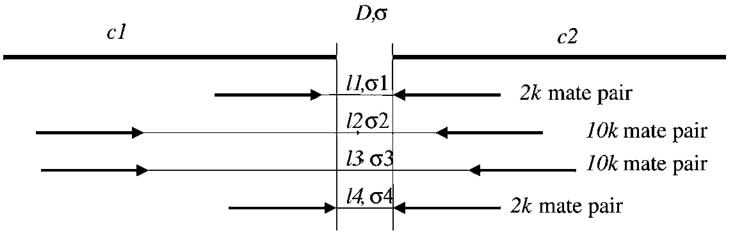


Figure 14 Here we depict two contigs that are linked by four mate-pairs. Each mate-pair provides an estimate (l_i, σ_i) of the gap between the two contigs, and simple statistics can be used to estimate a resulting mean distance D and standard deviation σ.

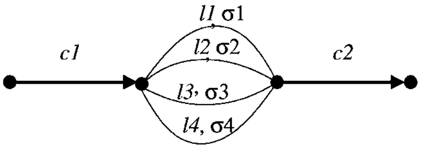


Figure 15 The two contigs and four mate-pairs shown in Figure 14 give rise to two contig-edges and four mate-edges in the contig-mate graph, as shown here.

the set of edges by a single *bundled mate-edge e*, whose mean length μ and standard deviation can be computed from the values for the original mate-edges, using straightforward statistics. Additionally, *e* is assigned a weight to reflect the number of mate-pairs that support it. Further edges can be bundled using so-called *transitive reduction*, which we do not describe here.

The goal of the *scaffolding* phase is to use the contig-mate graph to determine the true relative order and orientation of a set of contigs that are linked by mate-pairs. Most assemblers use different heuristics to address this problem.

We briefly discuss how this problem can be formalized (see Ref. [23] for details). An ordering or scaffolding of a set of contigs can be specified as a path *P* through the corresponding contig-mate graph. To this end, it may be necessary to infer "missing edges" between consecutive contig-edges in the path. To evaluate such a scaffolding one can look at all the mate-edges in the graph. We say that a mate-edge *e* is *satisfied* if the mate-pair layout implied by *e* is compatible with the ordering and orientation of contigs implied by *P*, otherwise *e* is called *unsatisfied*. Thus, the scaffolding phase can be stated as the following optimization problem: for a connected contig-mate graph, find a path *P* through the graph that contains all contig-edges, that possibly uses additional inferred edges and maximizes the number of satisfied mate-edges. This problem has been shown to be NP-hard [23].

Existing assemblers use straightforward heuristics in an attempt to form scaffolds. One heuristics that addresses the stated optimization problem directly is the "greedy path-merging" algorithm [23]. Given a connected contig-mate graph, the algorithm proceeds "bottom-up" as follows, maintaining a valid scaffolding $S \subseteq E$. Initially, all contig-edges $c_1, c_2, \ldots c_k$ are selected, but no others. At this stage, the graph consists of k selected paths $P_1 = (c_1), \ldots, P_k = (c_k)$. Then, in ordering of decreasing weight, each mate-edge $e = \{v, w\}$ is considered. If v and w lie in the same selected path P_i, then e is a chord of P_i and no action is necessary. If v and w are contained in two different paths P_i and P_j, we attempt to merge the two paths to obtain a new path P_k and accept such a merge, provided the increase of $S(G)$ (the number of satisfied mate-edges) is larger than the increase of $U(G)$ (the number of unsatisfied ones).

3.5 Computation of the Consensus Sequences

In a final step we need to determine the actual sequence for the target molecule. So far, we have discussed how to construct contigs, how to order them in scaffolds and how to address the problem of repeat resolution. The pairwise overlaps between reads provide only an approximate layout of the reads with respect to each other. To obtain a final layout, one needs to solve a special multiple alignment problem, with the following properties:

- The reads of the multiple alignment are almost identical.
- Quality values can be incorporated in the computation of the consensus sequence and be used to assign quality scores to consensus characters.
- The alignment is usually of depth 5–10 and very long (up to millions of base pairs).
- We need to compute the alignment very fast.

The fact that the initial read layout already gives an approximate alignment and the need to compute the alignment quickly results in the application of heuristics to solve the multiple alignment problem, since a generalization of the dynamic programming-based approach would result in a running time of $O(n^k)$ where k is about 5–10.

Most assemblers [4,26,50] implement the idea depicted in Figure 16 in some way. For each contig an alignment is "grown", starting with the pairwise alignment of the two left-most reads or the two reads with the best pairwise similarity. Then, the next read is aligned to the multiple alignment of the previous two reads and so on. Aligning a read to an alignment is usually done by converting the multiple alignment into a profile, and then employing an adaption of the pairwise alignment algorithm to the profile and the read (Figure 16). If quality values are at hand, they can be incorporated in to the alignment computation.

```
CGATAGCTAGG-CTAGCATCGC
      CTAGGGCTAGCATCGGGGCGCC
           GCTAGCAT-GGGGCGCCCTCGATCGTT
                 ATCG--GCGCCCTCG-TCGTTGCTAATAG
```

add next read to box

```
CGCCCTCGTCGTTGCTAATAGCGTTGCGC
```

Figure 16 Computation of the consensus sequence.

Once a multiple alignment has been determined, a consensus character is computed for each column of the multiple alignment. This can be done by simply voting on the majority character or, alternatively, by weighing the vote using the quality values [4]. If prior knowledge of the base composition is at hand, it can even be incorporated in a Bayesian approach [9] which computes the most likely consensus character and also derives a quality value for it. Some assembly programs employ *ad hoc* heuristics to incorporate the quality values [21] or use the approach implemented in the Phrap package [16,18,36]. Phrap avoids computing a multiple alignment altogether. Instead, it chooses a chain of single reads which are chosen such that they provide adjacent intervals of high-quality base calls. In each interval this single high-quality base is chosen for the consensus sequence.

Although this strategy does not use all available information, it avoids some artefacts introduced by the progressive method commonly used for the computation of multiple alignments. For example, Figure 17 shows a typical output of a progressive alignment on the left. Depending on the score function, the last read may result in three different alignments with other reads which are merged into a multi-alignment that introduces two Ts into the consensus sequence (depicted in blue). However, the multi-alignment on the right is more likely to be correct, since it can be explained with only two sequencing errors in the last read. Such additional characters are to be avoided, since they confound gene prediction algorithms and other

```
...AGCCGT--CTGA...  ...AGCCGT--CTGA.
...AGCCG-T-CTGA...  ...AGCCGT--CTGA.
...AGCCG-T-CTGA...  ...AGCCGT--CTGA.
...AGCCG--TCTGA...  ...AGCCGT--CTGA.
...AGCCGT--CTGA...  ...AGCCGT--CTGA.
...AGCCGTTTCTGA...  ...AGCCGTTTCTGA.
------------------  ----------------
...AGCCGTT-CTGA...  ...AGCCGT--CTGA.
incorrect           correct
```

Figure 17 Common error in consensus computation.

computational sequence analysis tools. A possible strategy to achieve this is to use iterative refinement strategies that correct such mistakes [2].

4 Examples of Existing Assemblers

In Table 1 we give an overview of current assemblers that are able to compute an assembly of large eukaryotic genomes and list some of the genomes that have been applied to. We use the attributes CBC and WGS to indicate whether the assembler follows more the CBC or the WGS paradigm. Assemblers that take clone information and a WGS data set as input are marked as hybrid.

Table 1 Recent assembly programs for eukaryotic genomes.

Name	Year	Strategy	Genomes (examples)
Celera	2000	WGS	*H. sapiens* [54], *M. musculus* [37], *D. melanogaster* [38], *Anopheles gambiae* [19],
GigAssembler	2001	CBC	*H. sapiens* [29]
ARACHNE	2002	WGS	*M. musculus* [4, 26]
JAZZ	2002	WGS	*Fugu rubripes*, *Ciona intestinalis* [3, 10]
RePS	2002	WGS	*Oryza sativa* [55, 57]
Barnacle	2003	CBC	*H. sapiens* [8]
PCAP	2003	WGS	*Caenorhabdtis briggsae*, *M. musculus* [20]
Phusion	2003	WGS	*M. musculus* [36]
Atlas	2004	hybrid	*Rattus norvegicus* [18, 46]

In the following we give short descriptions of the assemblers listed in Table 1. This will give the flavor of the latest algorithmic approaches and show that all assemblers use similar ideas.

4.1 The Celera Assembler

The Celera assembler was the first WGS assembler to assemble large eukaryotic genomes [38, 54]. It screens the reads, removes vector or linker sequence and keeps only the interval with an average sequence identity of 98%. The overlapper module compares all pairs of reads to detect high-fidelity overlaps. To avoid a quadratic number of overlap computations, the overlapper uses a k-mer index to exclude nonrelated pairs from the expensive overlap. This results in a read-overlap graph as described in Section 3.1. Regions of this graph are assembled into contigs whenever the initial arrangement of reads in this region is unique. The Celera assembler incorporates mate-pair information, and orders and orients the contigs. The remaining gaps are closed in a sequence of less and less conservative steps. First contigs are placed if they are "anchored" by two mate-pairs, then if they are anchored by one mate-pair and an overlap path and so on. For each contig a consensus sequence is computed based on a progressive multiple alignment and a local heuristics to remove merging artefacts.

4.2 The GigAssembler

The GigAssembler [29] was designed to assemble the human genome from the CBC data obtained from the HGP [25]. It had to assemble all BACs in a tiling layout of clones. In addition to this input set, it uses mate-pairs, mRNA and EST information to bridge gaps in scaffolds. The GigAssembler screens the input sequences for contaminations and masks known repeats. Additional sequence information (reads of mate-pairs, ESTs, mRNA and BAC end reads) is aligned to the input. Similar to the description in Section 3.1, GigAssembler builds a lookup table of 10-mers and then conducts a detailed alignment in regions where consecutive 10-mers match. The main routine of the GigAssembler builds sequence contigs (called "rafts") from overlapping initial sequence contigs within a clone. Then it builds clone contigs (called "barges") from overlapping clones, and orders and orients the resulting contigs into "supercontigs". These assemblies are combined into full chromosome assemblies.

4.3 The ARACHNE Assembler

The ARACHNE assembler was developed by a group at the MIT that was also a major partner in the HGP. In its first version [4] its functionality was tested by reassembling the genomes of *H. influenza*, *Saccharomyces cervisiae* and *D. melanogaster*, as well as the two smallest human chromosomes, 21 and 22. A later version of ARACHNE [26] was used in the public assembly of the mouse genome. ARACHNE appears to be modeled after the Celera assembler, with a few differences.

As a true WGS assembler its input consists of a set of reads and mate-pairs where the mate-pairs are taken from carefully length selected clone libraries. An overlap phase is conducted as outlined in Section 3.1. In addition, it employs an error correction phase using multiple alignments deduced from the pairwise overlaps.

The contig assembly phase differs from the one employed in the Celera assembler, since it directly incorporates mate-pairs by identifying "paired reads", which are reads of two mate-pairs where the two left and the two right reads overlap, respectively. This is a clever way to form contigs that are consistent in overlap and mate-pair information.

ARACHNE computes repeat boundaries by inspecting the pairwise overlaps. To detect remaining repetitive contigs ARACHNE uses an arrival statistic similar to the one described above, together with the fact that repetitive contigs are likely to have mate-pairs that link them in a contradictory way to other contigs.

ARACHNE uses mate-pairs to build scaffolds using a greedy algorithm that gives priority to merging contigs that are supported by the most links involving the shortest distance. This phase is followed by an attempt to fill gaps using contigs that were previously labeled as repetitive. Since the first labeling was conservative, this will often succeed. A consensus sequence is derived by heuristicly computing a multiple alignment.

4.4 The JAZZ Assembler

The JAZZ assembler is a modular assembler making use of different, sometimes already existing modules. The input reads are trimmed with respect to a window average of the quality values. In addition, they are checked for vector contamination which is then removed. The overlap phase is similar to the description of Section 3.1. An index of 16-mers is constructed. All 16-mers that occur too often are not used for triggering a more expensive alignment step. Then all reads that share more than ten non repetitive 16-mers are aligned using a banded Smith–Waterman algorithm. JAZZ constructs a scaffolded layout of reads. In particular, JAZZ postpones the computation of contigs until the consensus phase, which employs a consensus algorithm similar to Phrap. JAZZ tries to close gaps in scaffolds that are due to repeats in the genome.

4.5 The RePS Sssembler

The RePS uses also Phrap as its main assembly engine. It was primarily used to assemble the rice genome [55]. RePS masks out repeated 20-mers. The masked reads are handed to Phrap. As a post-Phrap step, RePS uses mate-pairs to fill gaps and build scaffolds. The strategies RePS uses are concepts borrowed from the Celera assembler.

4.6 The Barnacle Assembler

Barnacle [8] is an assembler that was used to reassemble the human genome from the public CBC data. In contrast to GigAssembler it does not make use of a physical map, but uses mate-pairs and clone data only (possibly augmented with chromosome assignments). Barnacle computes all pairwise local alignments of the input sequence. This is done using a strategy as described in Section 3.1. Using those overlaps, contigs in the input set are merged whenever possible, thereby reducing the number of contigs by an order of magnitude. The clone overlaps are deduced (two clones overlap if, and only if, at least one contig pair of the corresponding clones overlaps) and conflicts are resolved by heuristically enforcing the layout graph (clones are vertices, overlaps induce edges) to be an interval graph. Barnacle orients the contigs starting from the interval representation of clones and assign coordinates to sub-contigs. Again, possible inconsistencies result in the discarding of the contigs involved.

4.7 The PCAP Assembler

The PCAP assembler is a true WGS assembler that incorporates many aspects of the well known CAP3 assembler [21]. One interesting aspect of PCAP is the way repeats are identified *de novo* during the overlap computation. For this, the set of reads is partitioned into subsets that can be distributed on many computers. Then, in an iterative process, repeats are identified during the overlap computation and used to avoid the computation of repeat induced overlaps. To do this, some overlaps are computed and repetitive regions are identified based on those overlaps. These repeats are used in the next round of overlap computations, and so on.

The overlaps themselves are computed in a manner similar to the approach outlined in Section 3.1. Prior to the construction of contigs, the depth of coverage at every point in the initial layout determined by the overlaps is computed. Using the depth of coverage, overlaps are assigned a score that reflects whether they are repeat induced or not. Only overlaps that are likely to be unique are used in the contigging step. In addition, poor ends of reads are located and clipped and chimeric reads discarded (all based on pairwise overlaps).

Contigs are formed by inspecting read overlaps in decreasing order of their adjusted score. Then the CAP3 algorithm for scaffolding is applied. It consists of finding groups of mate-pairs that indicate a mis-assembly of the contig. If such mate-pairs can be found, the contig is corrected and the mate-pair consistency is checked again.

A simple gap filling strategy based on finding overlap paths is applied, multiple alignments are computed and a consensus sequence is derived as in CAP3. The computation of the consensus sequence involves a heuristic procedure that makes use of the quality values of the reads.

4.8 The Phusion Assembler

The Phusion assembler was primarily designed to assemble the mouse genome from a WGS data set at 7.5× coverage [36] and was developed in parallel with the ARACHNE assembler [4, 27]. It is a modular assembler in the sense that it incorporates an older program, i.e. Phrap, as an integral part of its operation. It screens the input reads for poor quality reads, which are completely removed, and conducts a screening for vector contamination.

Phusion computes a histogram of all k-mers for a suitable *k*. Similar to the Atlas assembler, it uses this histogram to exclude *k*-mers that occur too seldomly (probably sequencing errors) and too often (probably repeats). The remaining *k*-mers are used to group the reads into contiguous groups, which are then passed to Phrap for assembly. This strategy is quite similar to the Atlas assembler and the compartmentalized assembler used by Celera to assemble a version of the human genome [24].

Phusion uses Phrap as its assembly engine and iteratively computes assemblies of sets of reads. It checks the consistency of the mate-pairs in this set. Whenever an inconsistency is detected, Phusion splits the set and reassembles the parts using Phrap. This results in a number of contigs that might share sequence parts. Phusion tries to join contigs based on the number of shared reads and sequence overlaps, a strategy not unlike that of the GigAssembler [29]. The resulting, larger contigs are scaffolded, using the mate-pair information.

4.9 The Atlas Assembler

The Atlas assembly system is a suite of programs that form a hybrid assembler which uses reads from WGS and from CBC data sets. Thus, it is very similar to the compartmentalized assembler developed at Celera Genomics [24,54].

Atlas trims the input reads based on the error rate in a local window. It builds a *k*-mer index of the WGS reads, since these cover the genome uniformly. Similar to the Phusion assembler, it uses the fact that seldomly occurring *k*-mers are likely to contain sequencing errors, while abundant *k*-mers are likely to be repetitive. Atlas establishes the "rarity" of a *k*-mer in the overlap phase, using such *k*-mers to seed a banded alignment as described in Section 3.1.

The WGS reads are binned by using the localized BAC clone reads to "catch" the corresponding WGS reads. The reads in each BAC bin are assembled using Phrap. Since Phrap does not use mate-pairs during the assembly, the resulting contigs are checked for consistency and, if found to be inconsistent, split using the mate-pair information. The same information is then used to scaffold the resulting contigs. The improved BACs are called *eBACs*.

Atlas performs a meta-assembly of the eBACs. Based on overlap information and independent mapping data, a tiling path of eBACs is computed. The assembly induced by this tiling path is refined using *rolling-Phrap*, which is an iterative procedure calling Phrap in a window that is cleverly moved over the tiling path. The resulting large contigs are linked using mate-pairs and localized BAC reads, and then anchored on the genomic axis using external mapping data.

4.10 Other Assemblers

There are a number of other assemblers that are not described here, either because they have been outdated by more recent developments (e.g. Refs. [14, 50]) or because they have not been used to assemble large eukaryotic genomes. Specifically, we would like to mention the Euler (or Euler-DB) assembler [43], which formulates the assembly problem differently, using a *k*-mer graph.

The last years have seen the development of a host of different assembly programs, which nevertheless share a significant portion of algorithmic ideas. In general, sequence assembly can be seen as a concatenation of algorithmic modules with well-defined interfaces. Hence, we believe that it would be worthwhile to combine the best implementations of these modules, an approach that has been taken by the Amos consortium hosted by The Institute for Genomic Research (TIGR) [53].

5 Conclusion

Assembling whole eukaryotic genomes was deemed impossible only 15 years ago. Yet, an initiative was founded to tackle the seemingly gargantuan task of assembling the human genome. Whole-genome assembly of eukaryotic genomes, once strongly criticized as impractical, has now been successfully applied to a number of large genomes and has become the standard approach. This would not have been possible without bioinformatics support, the development of efficient assembly algorithms and solid engineering to implement those algorithms into robust computer programs that also handle all peculiarities of the data that are not captured in the mathematical models.

Acknowledgements. We would like to thank the Informatics Research group at Celera Genomics which, under the guidance of Gene Myers and Granger Sutton, developed the first large-scale whole genome shotgun assembler, and used it to assemble *D. melanogaster*, *H. sapiens*, and *M. musculus*. Another detailed and careful description of the shotgun assembly problem can be found in [49]. We would like to thank Gene Myers for reviewing an earlier version of the manuscript. His comments helped to greatly improve the chapter.

References

1 ALTSCHUL, S., W. GISH, W. MILLER, E. MYERS AND D. LIPMAN. 1990. A basic local alignment search tool. J. Mol. Biol. **215**: 403–10.

2 ANSON, E. AND E. MYERS. 1997. Realigner: a program for refining DNA sequence multialignments. J. Comput. Biol. **4**: 369–83.

3 APARICIO, S., J. CHAPMAN, E. STUPKA et al. 2002. Whole-genome shotgun assembly and analysis of the genome of *Fugu rubripes*. Science **297**: 1301–10.

4 BATZOGLOU, S., D. B. JAFFE, K. STANLEY, et al. 2002. ARACHNE: a whole genome shotgun assembler. Genome Res. **12**: 177–89.

5 BURKHARDT, S., A. CRAUSER, P. FERRAGINA, H.-P. LENHOF, E. RIVALS AND M. VINGRON. 1999. q-gram based database searching using suffix arrays. Proc. RECOMB **3**: 77–83.

6 CHANG, W. AND E. LAWLER. 1994. Sublinear approximate string matching. Algorithmica **12**: 327–44.

7 CHAO, K., W. PEARSON AND W. MILLER. 1992. Aligning two sequences within a specified diagonal band. Comput. App. Biosci. **8**: 481–7.

8 CHOI, V. AND M. FARACH-COLTON. 2003. Barnacle: an assembly algorithm for clone-based sequences of whole genomes. Gene **320**: 165–76.

9 CHURCHILL, G. AND M. S. WATERMAN. 1992. The accuracy of DNA sequences: estimating sequence quality. Genomics **14**: 89–98.

10 DEHAL, P., Y. SATOU, R. CAMPBELL, et al. 2002/2003. The draft genome of *Ciona intestinalis*: insights into chordate and vertebrate origins. Science **298**: 2157–68.

11 DEWAN, A., A. PARRADO AND T. MATISE. 2002. Map error reduction: using genetic and sequence-based physical maps to order closely linked markers. Human Hered. **54**: 34–44.

12 EWING, B., L. HILLIER, M. WENDL AND P. GREEN. 1998. Base-calling of automated sequencer traces using phred. I. Accuracy assessment. Genome Res. **8**: 175–85.

13 EWING, B. AND P. GREEN. 1998. Base-calling of automated sequencer traces using phred. II. Error probabilities. Genome Res. **8**: 186–94.

14 FLEISCHMANN, R. D., M. ADAMS, O. WHITE, et al. 1995. Whole-genome random sequencing and assembly of Haemophilus influenzae. Science **26**: 496–512.

15 GALLANT, J., D. MAIER AND J. STOERER. 1980. On finding minimal length superstrings. J. Comp. Syst. Sci. **20**: 50–58.

16 GREEN, P. 1994. PHRAP documentation. http: //www.phrap.org.

17 GREEN, P. 1997. Against a whole-genome shotgun. Genome Res. **7**: 410–7.

18 HAVLAK, P., R. CHEN, K. DURBIN, A. EGAN, Y. REN, X.-Z. SONG, G. WEINSTOCK AND R. GIBBS. 2004. The Atlas genome assembly system. Genome Res. **14**: 721–32.

19 HOLT, R. A., G. M. SUBRAMANIAN, A. HALPER, et al. 2002. A whole-genome assembly of *Drosophila*. Science **288**: 129–49.

20 HUANG, X., J. WANG, S. ALURU, S.-P. YANG AND L. HILLIER. 2003. PCAP: a whole-genome assembly program. Genome Res. **13**: 2164–70.

21 HUANG, X. AND A. MADAN. 1999. CAP3: A DNA sequence assembly program. Genome Res. **9**: 868–77.

22 HUANG, X. 1992. A contig assembly program based on sensitive detection of fragment overlaps. Genomics **14**: 18–25.

23 HUSON, D. H., K. REINERT AND E. W. MYERS. 2002. The greedy path-merging algorithm for contig scaffolding. J. ACM **49**: 603–15.

24 HUSON, D. H., K. REINERT, S. A. KRAVITZ, et al. 2001. Design of a compartmentalized shotgun assembler for the human genome. Bioinformatics **17**: 132–9.

25 INTERNATIONAL HUMAN GENOME SEQUENCING CONSORTIUM. 2001. Initial

sequencing and analysis of the human genome. Nature **409**: 860–921.

26 JAFFE, D., J. BUTLER, S. GNERRE, et al. 2003. Whole-genome sequence assembly for mammalian genomes: ARACHNE 2. Genome Res. **13**: 91–6.

27 JAFFE, D., J. BUTLER, S. GNERRE, et al. 2003. Whole genome sequence assembly for mammalian genomes: ARACHNE 2. Genome Res. **13**: 91–6.

28 KECECIOGLU, J. AND J. YU. 2001. Separating repeats in DNA sequence assembly. Proc. RECOMB **5**: 176–83.

29 KENT, W. J. AND D. HAUSSLER. 2001. Assembly of the working draft of the human genome with GigAssembler. Genome Res. **11**: 1541–8.

30 KEVLES, D. J. AND L. HOOD. 1992. *The Code of Codes: Scientific and Social Issues in the Human Genome Project*. Harvard University Press, Cambridge, MA.

31 LANDER, E. S. AND M. S. WATERMAN. 1988. Genomic mapping by fingerprinting random clones: a mathematical analysis. Genomics **2**: 231–9.

32 MARRA, M., L. HILLIER AND R. H. WATERSTON. 1998. Expressed sequence tags – ESTablishing bridges between genomes. Trends Genet. **14**: 4–7.

33 MELDRUM, D. 2000. Automation for genomics, part 1: preparation for sequencing. Genome Res. **10**: 1081–92.

34 MELDRUM, D. 2000. Automation for genomics, part 2: sequencers, microarrays, and future trends. Genome Res. **10**: 1288–303.

35 MEWES, W., K. ALBERMANN, M. BÄHR, et al. 1997. An overview of the yeast genome. Nature **387**: 7–8.

36 MULLIKIN, J. AND Z. NING. 2003. The Phusion assembler. Genome Res. **13**: 81–90.

37 MURAL, R. J., M. D. ADAMS, G. W. MYERS, et al. 2002. A comparison of whole-genome shotgun-derived mouse chromosome 16 and the human genome. Science **296**: 1661–71.

38 MYERS, E. W., G. G. SUTTON, A. L. DELCHER, et al. 2000. A whole-genome assembly of *Drosophila*. Science **287**: 2196–204.

39 MYERS, E. 1990. A fast bit-vector algorithm for approximate string matching based on dynamic programming. J. ACM **46**: 495–515.

40 MYERS, E. 1994. A sublinear algorithm for approximate keyword matching. Algorithmica **12**: 345–74.

41 MYERS, E. 1995. Toward simplifying and accurately formulating fragment assembly. J. Comput. Biol. **2**: 275–90.

42 PALCA, J. 1986. Human genome – Department of Energy on the map. Nature **321**: 371.

43 PEVZNER, P., H. TANG AND M. WATERMAN. 2001. An eulerian path approach to DNA fragment assembly. Proc. Natl Acad. Sci. USA **98**: 9748–53.

44 PEVZNER, P. 2000. *Computational Molecular Biology.* MIT Press, Cambridge, MA.

45 RASMUSSEN, K., J. STOYE AND E. W. MYERS. 2005. Efficient q-gram filters for finding all *epsilon*-matches over a given length. Proc. RECOMB **9**: 189–203.

46 RAT GENOME SEQUENCING CONSORTIUM. 2004. The genome sequence of the brown Norway rat yields insights into mammalian evolution. Nature **428**: 493–521.

47 SCHULER, G., M. BOGUSKI, E. STEWART, et al. 1996. A gene map of the human genome. Science **274**: 540–6.

48 SINSHEIMER, A. L. 1985. The Santa Cruz Workshop – May 1985. Genomics **5**: 954–6.

49 SUTTON, G. AND I. DEW. 2005. Shotgun fragment assembly. In *Series in Systems Biology.* Oxford University Press, Oxford: to appear.

50 SUTTON, G. G., O. WHITE, M. D. ADAMS AND A. R. KERLAVAGE. 1995. TIGR Assembler: A new tool for assembling large shotgun sequencing projects. Genome Sci. Technol. **1**: 9–19.

51 TAMMI, M., E. ARNER, E. KINDLUND AND B. ANDERSSON. 2003. Correcting errors in shotgun sequences. Nucl. Acids Res. **31**: 4663–72.

52 TAMMI, M., E. ARNER, T. BRITTON AND B. ANDERSSON. 2002. Separation of nearly identical repeats in shotgun assemblies using defined nucleotide positions, DNPs. Bioinformatics **18**: 379–88.

53 TIGR. 2005. The AMOS web page, http://www.tigr.org/software/AMOS.

54 VENTER, J. C., M. D. ADAMS, E. W. MYERS, et al., 2001. The sequence of the human genome. Science **291**: 1145–1434.

55 WANG, J., G. K.-S. WONG, P. NI, et al., 2002. RePS: a sequence assembler that masks exact repeats identified from the shotgun data. Genome Res. **12**: 824–31.

56 WEBBER, J. L. AND E. W. MYERS. 1997. Human whole-genome shotgun sequencing. Genome Res. **7**: 401–9.

57 YU, J., S. HU, J. WANG, et al. 2002. A draft sequence of the rice genome (*Oryza sative* L. ssp. *indica*). Science **296**: 79–92.

Part 3 Sequence Analysis

3 Sequence Alignment and Sequence Database Search

Martin Vingron

1 Introduction

In evolutionary studies two characters are called homologous when they share common evolutionary ancestry. Genes may also be homologous, which usually is reflected by similarity among their DNA or amino acid sequences. Furthermore, homology among genes frequently implies that they are functionally similar. Thus, there are two good reasons to compare the sequences of genes or proteins, i.e. the unraveling of evolutionary relationships and extrapolating function from one gene to another.

The basis for the study of sequence similarity is the comparison of two sequences which will be dealt with in Section 2. Sequence comparisons are performed in large numbers when searching sequence databases for sequences that are similar to a query sequence. Algorithms for this purpose need to be fast, even at the expense of sensitivity. Section 3 discusses the widely used heuristic approaches to database searching. However, the algorithms we are designing for the purpose of quantifying sequence similarity can only be as good as our understanding of evolutionary processes and thus they are far from perfect. Therefore, results of algorithms need to be subjected to a critical test using statistics. Methods for the assessment of the statistical significance of a finding are introduced in Section 4.

Genes do not come in pairs, but rather in large families. Consequently, the need arises to align more than two sequences at a time, which is done by multiple alignment programs. Computationally a very hard problem, it has attracted considerable attention from the area of algorithm development. Section 5 presents the basic approaches to multiple sequence alignment.

Bioinformatics – From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

Section 6 builds on the knowledge of a multiple alignment and introduces how to exploit the information contained in several related sequences for the purpose of identifying additional related sequences in a database. The last section covers methods and resources to structure the entire space of protein sequences.

2 Pairwise Sequence Comparison

2.1 Dot plots

Dot plots are probably the simplest way of comparing sequences [55]. A dot plot is a visual representation of the similarities between two sequences. Each axis of a rectangular array represents one of the two sequences to be compared. A window length is fixed, together with a criterion under which two sequence windows are deemed to be similar. A typical choice for this similarity criterion would be a certain fraction of matching residues within a window. Whenever one window in one sequence resembles another window in the other sequence, a dot or short diagonal is drawn at the corresponding position of the array. Thus, when two sequences share similarity over their entire length a diagonal line extends from one corner of the dot plot to the diagonally opposite corner. If two sequences only share patches of similarity this is revealed by diagonal stretches.

Figure 1 shows an example of a dot plot. There, the coding DNA sequences of the α- and β-chains of human hemoglobin are compared to each other. For this computation the window length was set to 31. The program adds up the matches within a window and the gray value at the position corresponding to the center of the window is set according to the quality of the match at that position. One can clearly discern a diagonal trace along the entire length of the two sequences. Note the jumps where this trace changes to another diagonal of the array. These jumps correspond to the position where one sequence has more (or fewer) letters than the other one. Figure 1 was produced using the program "dotter" [71].

Dot plots are a powerful method of comparing two sequences. They do not predispose the analysis in any way such that they constitute the ideal first-pass analysis method. Based on the dot plot the user can decide whether they deal with a case of global, i.e. beginning-to-end, similarity or local similarity. "Local similarity" denotes the existence of similar regions between two sequences that are embedded in the overall sequences which lack similarity. Sequences may contain regions of self-similarity which are frequently termed internal repeats. A dot plot comparison of the sequence itself will reveal internal repeats by displaying several parallel diagonals (see also Chapter 7).

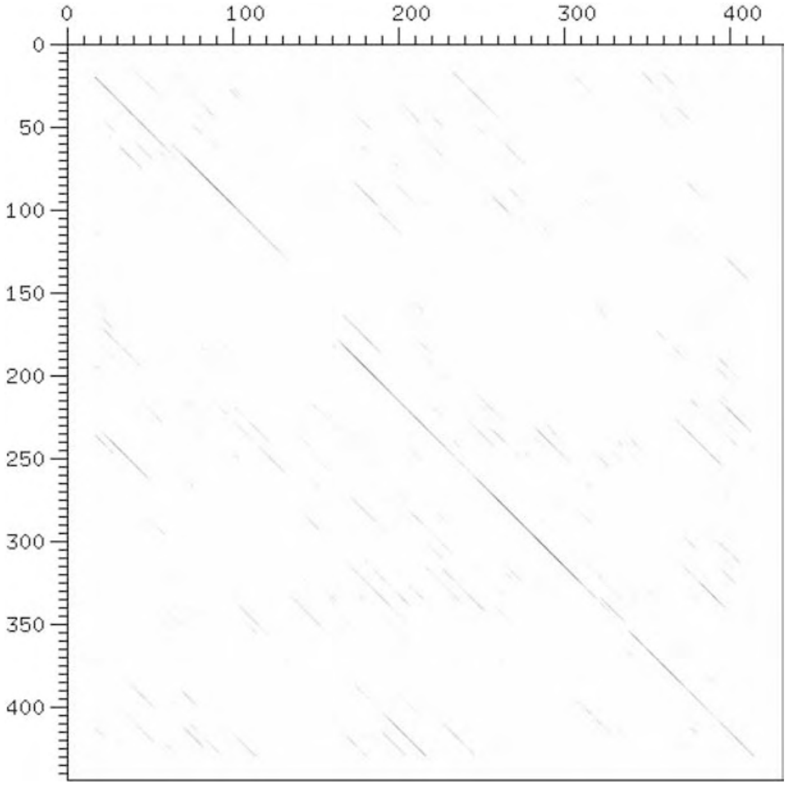
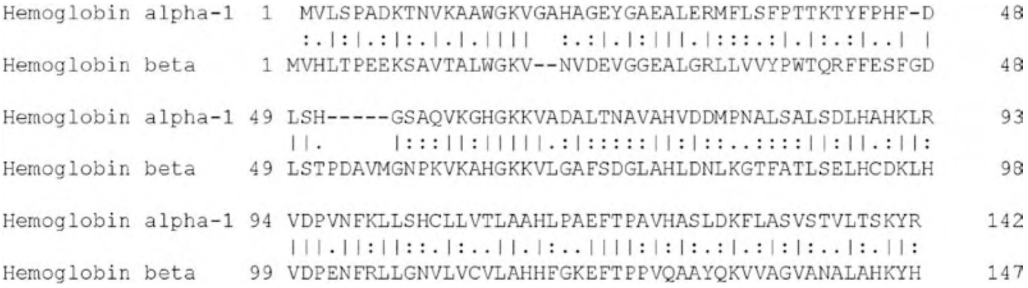


Figure 1 Dot plot comparing two hemoglobin sequences. The horizontal axis corresponds to the sequence of the human β-hemoglobin chain; the vertical sequence (numbered from top to bottom) represents the human α-hemoglobin chain.

Instead of simply deciding if two windows are similar, a quality function may be defined. In the simplest case, this could be the number of matches in the window. For amino acid sequences the physical relatedness between amino acids may give rise to a quantification of the similarity of two windows. For example, when a similarity matrix on the amino acids (like the Dayhoff matrix, see below) is used one might sum up these values along the window. However, when this similarity matrix contains different values for exact matches this leads to exactly matching windows of different quality. The dot plot method of Argos [5] is an intricate design that reflects the physical relatedness of amino acids. The program dotter [71] is an X-windows-based program that allows for displaying dot plots for DNA, for proteins and for comparison of DNA to protein.



```
Hemoglobin alpha-1  1  MVLSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-D     48
                       :.|:|.:|:.|.|.||||  :.:|.|:|||.|:::.:|.|:.:|..| |
Hemoglobin beta     1 MVHLTPEEKSAVTALWGKV--NVDEVGGEALGRLLVVYPWTQRFFESFGD     48

Hemoglobin alpha-1 49 LSH-----GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKLR     93
                      ||.     |:::||:|||||.:|:::::||:|::..::::||:||.:||:
Hemoglobin beta    49 LSTPDAVMGNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKLH     98

Hemoglobin alpha-1 94 VDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR    142
                      |||.||:||::.|:..||.|:..||||:|:|:.:|.:|:|:..|:.||:
Hemoglobin beta    99 VDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH    147
```

Figure 2 Sequence alignment between the amino acid sequences of human hemoglobin α- and β-chains. Note that these are the same genes for which the dot plot of the corresponding coding DNA sequences is shown in Figure 1.

2.2 Sequence Alignment

A sequence alignment [81] is a scheme of writing one sequence on top of another such that the residues in the same position are deemed to have a common evolutionary origin. If the same letter occurs at the same position in both sequences, then this position has been conserved in evolution (or, coincidentally, mutations from another ancestral residue have given rise to the same letter twice). If the letters differ it is assumed that both derive from the same ancestral letter, which could be one of the two or neither. Homologous sequences may have different length, though, which is generally explained through insertions or deletions in sequences. Thus, a letter or a stretch of letters may be paired up with dashes in the other sequence to signify such an insertion or deletion. Since an insertion in one sequence can always be seen as a deletion in the other one sometimes uses the term "indel" (or, simply, "gap"). Figure 2 depicts an example of an alignment. The sequences aligned there are the proteins derived from the coding sequences compared in Figure 1. Note that the first stretch of contiguously aligned amino acids (up to the WGKV match) corresponds to the first diagonal stretch in the dot plot of Figure 1. The subsequent insertion of 2 amino acids in the α-chain corresponds to linking this first diagonal to the second one, which is located around position 100. Likewise, the next five-letter gap in the alignment corresponds to the join from the second diagonal to the third, starting around position 200 in the dot plot.

In such a simple evolutionarily motivated scheme, an alignment mediates the definition of a distance for two sequences. One generally assigns a score of zero to a match, some positive number to a mismatch and a larger positive number to an indel. By adding these values along an alignment one obtains a score for this alignment. A distance function for two sequences can be defined by looking for the alignment which yields the minimum score.

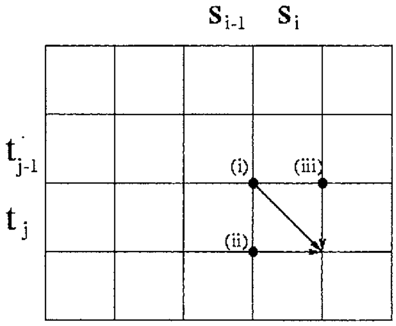


Figure 3 Schematic representation of the edit matrix comparing two sequences. The arrows indicate how an alignment may end according to the three cases described in the text.

Naively, the alignment that realizes the minimal distance between two sequences could be identified by testing all possible alignments. This number, however, is prohibitively large; luckily, using dynamic programming, the minimization can be effected without explicitly enumerating all possible alignments of two sequences. To describe this algorithm [64] denote the two sequences by $s = s_1, \ldots, s_n$ and $t = t_1, \ldots, t_m$. The key to the dynamic programming algorithm is the realization that for the construction of an optimal alignment between two stretches of sequence $s_1, \ldots, s_i$ and $t_1, \ldots, t_j$ it suffices to inspect the following three alternatives:

(i) The optimal alignment of $s_1, \ldots, s_{i-1}$ with $t_1, \ldots, t_{j-1}$, extended by the match between s_i and t_j;

(ii) The optimal alignment of $s_1, \ldots, s_{i-1}$ with $t_1, \ldots, t_j$, extended by matching s_i with a gap character "–";

(iii) The optimal alignment of $s_1, \ldots, s_i$ with $t_1, \ldots, t_{j-1}$, extended by matching a gap character "–" with t_j.

Each of these cases also defines a score for the resulting alignment. This score is made up of the score of the alignment of the so far unaligned sequences that used plus the cost of extending this alignment. In case (i), this cost is determined by whether or not the two letters are identical; in cases (ii) and (iii), the cost of extension is the penalty assigned to a gap. The winning alternative will be the one with the best score (Figure 3).

To implement this computation one fills in a matrix the axes of which are annotated with the two sequences s and t. It is helpful to use north, south, west and east to denote the sides of the matrix. Let the first sequence extend from west to east on the north side of the matrix. The second sequence extends from north to south on the west side of the matrix. We want to fill the matrix starting in the north-western corner, working our way southward row by row, filling each row from west to east. To start, one initializes the northern and western margin of the matrix, typically with gap penalty values. After this initialization the above rules can be applied. A cell (i, j) that is already filled contains the score of the optimal alignment of the sequence $s_1, \ldots, s_i$ with $t_1, \ldots, t_j$. The score of each such cell can be determined by inspecting the cell immediately north-west of it [case (i)], the cell west [case (ii)] and the one north [case (iii)] of it, and deciding for the best scoring option. When the procedure reaches the south-eastern corner, that last cell contains the score of the best alignment. The alignment itself can be recovered as one backtracks from this cell to the beginning, each time selecting the path that had given rise to the best option.

The idea of assigning a score to an alignment and then minimizing or maximizing over all alignments is at the heart of all biological sequence alignment. However, many more considerations have influenced the definition of the scores and made sequence alignment applicable to a wide range of biological settings. First, note that one may either define a distance or a similarity function of an alignment. The difference lies in the interpretation of the values. A distance function defines positive values for mismatches or gaps and then aims at minimizing this distance. A similarity function assigns high values to matches and low values to gaps, and then maximizes the resulting score. The basic structure of the algorithm is the same for both cases. In 1981, Smith and Waterman [69] showed that for global alignment, i.e. when a score is computed over the entire length of both sequences, the two concepts are in fact equivalent. Thus, it is now customary to choose the setting that gives more freedom for appropriately modeling the biological question of interest.

In the similarity framework one can easily distinguish among the different possible mismatches and also among different kinds of matches. For example, a match between two tryptophans is usually regarded to be more important than a match between two alanines. Likewise, the pairing of two hydrophobic amino acids like leucine and isoleucine is preferable to the pairing of a hydrophobic with a hydrophilic residue. Scores are used to describe these similarities and are usually represented in the form of a symmetric 20×20 matrix, assigning a similarity score to each pair of amino acids. Although easy to understand from the physical characteristics of the amino acids, the values in such a matrix are usually derived based on an evolutionary model that enables one to estimate whether particular substitutions are preferred or avoided. To be more precise, the similarity score for 2 amino acids is defined as the logarithm of the likelihood ratio of the two residues being homologous versus finding them at their corresponding positions due to chance. This approach has been pioneered by Dayhoff [17] who computed a series of amino acid similarity matrices. Each matrix in this series corresponds to a particular evolutionary distance among sequences. This distance is measured in a unit called 1 PAM, for 1 Accepted Point Mutation (in 100 positions). The matrices carry names like PAM120 or PAM250, and are supposed to be characteristic for evolutionary distances of 120 or 250 PAM, respectively. Other more recent series of matrices are the BLOSUM matrices [27] or the VT series of matrices [57]. For every matrix one needs to find appropriate penalties for gaps.

The treatment of gaps deserves special care. The famous algorithm by Needleman and Wunsch [60] did not impose any restrictions on the penalty assigned to a gap of a certain length. For reasons of computational speed, later gap penalties were restricted to a cost function linear in the number of deleted (inserted) residues [64]. This amounts to penalizing every single indel. However, since a single indel tends to be penalized such that it is considerably inferior to a mismatch, this choice resulted in longer gaps being quite expensive and thus unrealistically rare. As a remedy, one mostly uses a gap penalty function which charges a *gap open penalty* for every gap that is introduced and penalizes the length with a *gap extension penalty* which is charged for every inserted or deleted letter in that gap. Clearly, this results in an affine linear function in the gap length, frequently written as $g(k) = a + b * k$ [80].

With the variant of the dynamic programming algorithm first published by Gotoh [23] it became possible to compute optimal alignments with affine linear gap penalties in time proportional to the product of the lengths of the two sequences to be aligned. This afforded a speed-up by an order of magnitude compared to a naive algorithm using the more general gap function. A further breakthrough in alignment algorithms development was provided by an algorithm that could compute an optimal alignment using computer memory only proportional to the length of one sequence instead of their product. This algorithm by Myers and Miller [59] is based on work by Hirshberg [29].

Depending on the biological setting, several kinds of alignment are in use. When sequences are expected to share similarity extending from the beginning of the sequences to their ends, they are aligned globally. This means that each residue of either sequence is part either of a residue pair or a gap. In particular, it implies that gaps at the ends are charged like any other gap. This, however, is a particularly unrealistic feature of a global alignment. While sequences may very well share similarity over their entire length (see the example dot plot of two hemoglobin chains in Figure 1), their respective N- and C-termini usually are difficult to match up, and differences in length at the ends are more of a rule than an exception. Consequently, one prefers to leave gaps at the ends of the sequences unpenalized. This variant is easy to implement in the dynamic programming algorithm. Two modifications are required. First, the initialization of the matrix needs to reflect the gap cost of zero in the margin of the matrix. Second, upon backtracking, one does not necessarily start in the corner of the matrix, but rather searches the margins for the maximum from which to start. Variants of this algorithm that penalize only particular end-gaps are easy to derive and can be used, for example, to fit one sequence into another or to overlap the end of one sequence with the start of another.

In many cases, however, sequences share only a limited region of similarity. This may be a common domain or simply a short region of recognizable similarity. This case is dealt with by so-called local alignment in an algorithm due to Smith and Waterman [69]. Local alignment aims at identifying the best pair of regions, one from each sequence, such that the optimal (global) alignment of these two regions is the best possible. This relies on a scoring scheme that maximizes a similarity score because otherwise an empty alignment would always yield the smallest distance. Naively, the algorithm to compute a local alignment would need to inspect every pair of regions and apply a global alignment algorithm to it. The critical idea of Smith and Waterman was to offer the maximization in each cell of the matrix a fourth alternative: a zero to signify the beginning of a new alignment. After filling the dynamic programming matrix according to this scheme, backtracking starts from the cell in the matrix that contains the largest value.

Upon comparing a dot plot and a local alignment one might notice regions of similarity visible in the dot plot, but missing in the alignment. While in many cases there exist gap penalty settings that would include all interesting matching regions in the alignment, generally it requires the comparison with the dot plot to notice possible misses. This problem is remedied by an algorithm due to Waterman and Eggert [82] which computes suboptimal, local and nonoverlapping alignments. It starts with the application of the Smith–Waterman algorithm, i.e. a dynamic programming matrix is filled and backtracking from the matrix cell with the largest entry yields the best local alignment. Then the algorithm proceeds to delineate a second-best local alignment. Note that this cannot be obtained by backtracking from the second-best matrix cell. Such an approach would yield an alignment largely overlapping the first one and thus containing little new information. Instead, those cells in the dynamic programming matrix are set to zero from where backtracking would lead into the prior alignment. This can be regarded as "resetting" the dynamic programming matrix after having deleted the first alignment. Then the second best alignment is identified by looking for the maximal cell in the new matrix and starting backtracking from there. Iteration of this procedure yields one alternative, nonoverlapping alignment after the other in order of descending quality. Application of this algorithm avoids possibly missing matching regions because even under strong gap penalties the procedure will eventually show all matching regions.

There is an interesting interplay between parameters, particularly the gap penalty, and the algorithmic variant used. Consider a pair of sequences whose similar regions can in principle be strung together into an alignment. Under a weak gap penalty the Smith–Waterman algorithm has a chance to identify this entire alignment. On the other hand, not knowing about the similarity between the sequences ahead of time, a weak gap penalty might also yield all kinds of spurious aligned regions. The Waterman–Eggert algorithm is a valid alternative. The gap penalty can be chosen fairly stringently. The first (i.e. the Smith–Waterman) alignment will then identify only the best-matching region out of all the similar regions. By iterating the procedure, though, this algorithm will successively identify the other similar regions as well. For a detailed discussion of these issues, see Vingron and Waterman [79].

3 Database Searching I: Single-sequence Heuristic Algorithms

This section takes a first look at the problem of identifying those sequences in a sequence database that are similar to a given sequence. This task arises, for example, when a gene has been newly sequenced and one wants to determine whether a related sequence already exists in a database. Generally, two settings can be distinguished. The starting point for the search may either be a single sequence, with the goal of identifying its relatives, or a family of sequences, with the goal of identifying further members of that family. Searching through a database needs to be fast and sensitive, but the two objectives contradict each other. Fast methods have been developed primarily for searching with a single sequence and this will be the topic of this section.

When searching a database with a newly determined DNA or amino acid sequence – the so-called query sequence – the user typically lacks knowledge of whether an expected similarity might span the entire query or just part of it. Likewise, they will be ignorant of whether the match will extend along the full length of some database sequence or only part of it. Therefore, one needs to look for a local alignment between the query and any sequence in the database. This immediately suggests the application of the Smith–Waterman algorithm to each database sequence. One should take care, though, to apply a fairly stringent gap penalty such that the algorithm focuses on the regions that really match. After sorting the resulting scores the top scoring database sequences are the candidates of interest.

Several implementations of this procedure are available, most prominently the SSEARCH program from the FASTA package [63]. There exist implementations of the Smith–Waterman algorithm that are tuned for speed like one using special processor instructions [85] and, among others, one by Barton [9].

Depending on implementation, computer and database size, a search with such a program takes on the order of 1 min.

The motivation behind the development of other database search programs has been to emulate the Smith–Waterman algorithm's ability to discern related sequences while at the same time performing the job in much less time. To this end, one usually makes the assumption that any good alignment that one wishes to identify contains, in particular, some stretch of ungapped similarity. Furthermore, this stretch will tend to contain a certain number of identically matching residues and not only conservative replacements. Based on these assumptions, most heuristic programs rely on identifying a well-matching core and then extending it or combining several of these. With hindsight, the different developments in this area can further be classified according to a traditional distinction in computer science by which one either preprocesses the query or the text (i.e. the database). Preprocessing means that the string is represented in a different form that allows for faster answers to particular questions, e.g. whether the string contains a certain subword.

The FASTA program (part of a package [63] that usually goes by the same name) sets a size k for k-tuple subwords. For DNA sequences, the parameter k might typically be set to 7, while for amino acid sequences 2 would be a reasonable choice. The program then looks for diagonals in the comparison matrix between the query and search sequence along which many k-tuples match. This can be done very quickly based on a preprocessed list of k-tuples contained in the query sequence. The set of k-tuples can be identified with an array whose length corresponds to the number of possible tuples of size k. This array is linked to the indices of the positions at which the particular k-tuples occur in the query sequence. Note that a matching k-tuple at index i in the query and at index j in the database sequence can be attributed to a diagonal by subtracting one index from the other. Therefore, when inspecting a new sequence for similarity one walks along this sequence inspecting each k-tuple. For each of them one looks up the indices of the positions at which it occurs in the query, computes the index-difference to identify the diagonal and increases a counter for this diagonal. After inspecting the search sequence in this way a diagonal with a high count is likely to contain a well-matching region. In terms of the execution time, this procedure is only linear in the length of the database sequence and can easily be iterated for a whole database. Of course this rough outline needs to be adapted to focus on regions where the match density is high and link nearby, good diagonals into alignments.

The other widely used program to search a database is called BLAST [1,3]. BLAST follows a similar scheme in that it relies on a core similarity, although with less emphasis on the occurrence of exact matches. This program also aims at identifying core similarities for later extension. The core similarity is defined by a window with a certain match density on DNA or with an amino acid similarity score above some threshold for proteins. Independent of the exact definition of the core similarity, BLAST rests on the precomputation of all strings which are similar in the given sense to any position in the query. The resulting list may contain on the order of 1000 or more words, each of which if detected in a database gives rise to a core similarity. In BLAST nomenclature this set of strings is called the neighborhood of the query. In fact, the code to generate this neighborhood is exceedingly fast.

Given the neighborhood, a finite automaton is used to detect occurrences in the database of any string from the neighborhood. This automaton is a program constructed "on the fly" and specifically for the particular word neighborhood that has been computed for a query. Upon reading through a database of sequences, the automaton is given an additional letter at a time and decides whether the string that ends in this letter is part of the neighborhood. If so, BLAST attempts to extend the similarity around the neighborhood and if this is successful reports a match.

As with FASTA, BLAST has also been adapted to connect good diagonals and report local alignments with gaps. BLAST converts the database file into its own format to allow for faster reading. This makes it somewhat unwieldy to use in a local installation unless someone takes care of the installation. FASTA, however, is slower, but easier to use. There exist excellent web servers that offer these programs, in particular at the National Center for Biotechnology Information [43] and at the European Bioinformatics Institute [41] where BLAST or FASTA can be used on up-to-date DNA and protein databases.

According to the above-mentioned distinction among search methods into those that preprocess the pattern and those that preprocess the text, there also is the option of transforming a DNA or amino acid database such that it becomes easier to search. This route was taken, for example, by a group from IBM developing the FLASH [14] program. They devised an intricate, although supposedly very space-consuming technique of transforming the database into an index for storing the offsets of gapped k-tuples. The QUASAR program by Burkhard and coworkers [13] preprocesses the database into a so-called suffix array, similar to a suffix tree, yet simple to keep on disk. Programs in practical use for quickly searching entire genomes are BLAT [50] and SSAHA [61].

With the availability of expressed sequence tags (ESTs) it has become very important to match DNA sequence with protein sequence in such a way that a possible translation can be maintained throughout the alignment. Both FASTA and BLAST packages contain programs for this and related tasks. When coding DNA is compared to proteins, gaps are inserted in such a way as to maintain a reading frame. Likewise, a protein sequence can be searched versus a DNA sequence database. The search of DNA versus DNA with an emphasis on matching regions that allow for a contiguous translation is not so well supported. Although a dynamic programming algorithm for this task is feasible, the existing implementation in BLAST compares all reading frames.

4 Alignment and Search Statistics

Alignment score is the product of an optimization, mostly a maximization procedure. As such it tends to be a large number sometimes suggesting biological relatedness where there is none. In pairwise comparisons the user still has a chance to study an alignment by eye in order to judge its validity; however, upon searching an entire database automatic methods are necessary to attribute a statistical significance to an alignment score.

In the early days of sequence alignment, the statistical significance of the score of a given pairwise alignment was assessed using the following procedure. The letters of the sequences are permuted randomly and a new alignment score is calculated. This is repeated roughly 100 times, and the mean and standard deviation of this sample are calculated. The significance of the given alignment score is reported in "number of standard deviations above the mean", also called the Z-score. Studying large numbers of random alignments is correct, in principle. However, the significance of the alignment should then be reported as the fraction of random alignments that score better than the given alignment. The procedure described assumes that these scores are distributed normally. Since the random variable under study – the score of an optimal alignment – is the maximum over a large number of values, this is not a reasonable assumption. In fact, the lack of fit quickly becomes obvious when trying to fit a normal distribution to the data. The second argument against this way of calculating significance is a pragmatic one: the procedure needs to be repeated for every alignment under study because the effect of the sequence length cannot be accounted for.

Based on the work of several researchers [48, 70], it has meanwhile become apparent that alignment score as well as scores from database searches obey a so-called extreme-value distribution. This is not surprising given that extreme-value distributions typically describe random variables that are the result of maximization. In sequence alignment, there are analytical results confirming the asymptotic convergence to an extreme-value distribution for the case of local alignment without gaps, i.e. the score of the best-matching contiguous diagonal in a comparison [18]. This is also a valid approximation to the type of matching effected in the database search program BLAST. Thus, this approach has become widely used and, in fact, has contributed significantly to the popularity of database search programs because significance measures have made the results of the search much easier to interpret.

The statistical significance of an event like observing a sequence alignment of a certain quality is the probability to observe a better value as a result of chance alone. This quantity is refereed to as the p-value. For example, a p-value of 10^{-3} is interpreted as expecting to see an excess of the given threshold in one in a 1000 experiments. To compute this one needs to model chance alignments, which is precisely what the statistician means by deriving the distribution of a random variable. The probability that a chance result would exceed an actually obtained threshold S is 1 minus the value of the cumulative distribution function evaluated at that threshold. In sequence alignment, this cumulative distribution function is generally expressed as [48]:

$$\exp(mnKe^{\lambda S})$$

where m and n are the lengths of the sequences compared, and K and λ are parameters which need to be computed (where possible) or derived by simulation. K and λ depend on the scoring matrix used (e.g. the PAM120 matrix) and the distribution of residues. Hence, for any scoring system these parameters are computed beforehand and the statistical significance of an alignment score S is then computed by evaluating the formula with the length of the two sequences compared.

The most prominent case for which the parameters K and λ can be defined analytically is local alignment without gaps. Algorithmically this amounts to computing a Smith–Waterman alignment under very high gap penalties such that the resulting alignment will simply not contain any gaps. Since this notion of alignment also guides the heuristic used by the BLAST database search program, the resulting statistical estimates are primarily used in database searching. In this application, one of the lengths is the length of the input sequence and the other length can be chosen on the order of the length of the concatenated sequences from the database that is being searched. Alternatively, one can think of the database search as a repetition of many individual pairwise comparisons, which amounts to repeating the experiment "sequence comparison" many times. In this setting, the number of false positives one expects to find can be determined as the product of the p-value of the individual comparison and the number of times the experiment is repeated, i.e. the number of sequences in the database. This expected number of false positives is referred to as the E-value. A typical E-value threshold for a database search would be, for example, 1, indicating that the score cutoff is chosen such that among the sequences faring better than the cutoff one expects to find one false-positive hit.

When gaps are allowed, the determination of K and λ is more complex because an approximation of the distribution function of alignment score by an extreme-value distribution as above is not always valid. Generally speaking, it is allowed only for sufficiently strong gap penalties where alignments remain compact as opposed to spanning the entire sequences. Under sufficiently strong gap penalties, though, it has been demonstrated that the approximation is indeed valid just like for infinite gap penalties [79]. However, it is not possible any more to compute the values of the parameters K and λ analytically. As a remedy one applies simulations in which many alignments of randomly generated sequences are computed and the parameters are determined based on fitting the empirical distribution function with an extreme-value distribution [83]. As in the case above, this procedure allows for determining parameters beforehand and computing significance by putting the lengths of the sequences into the formula.

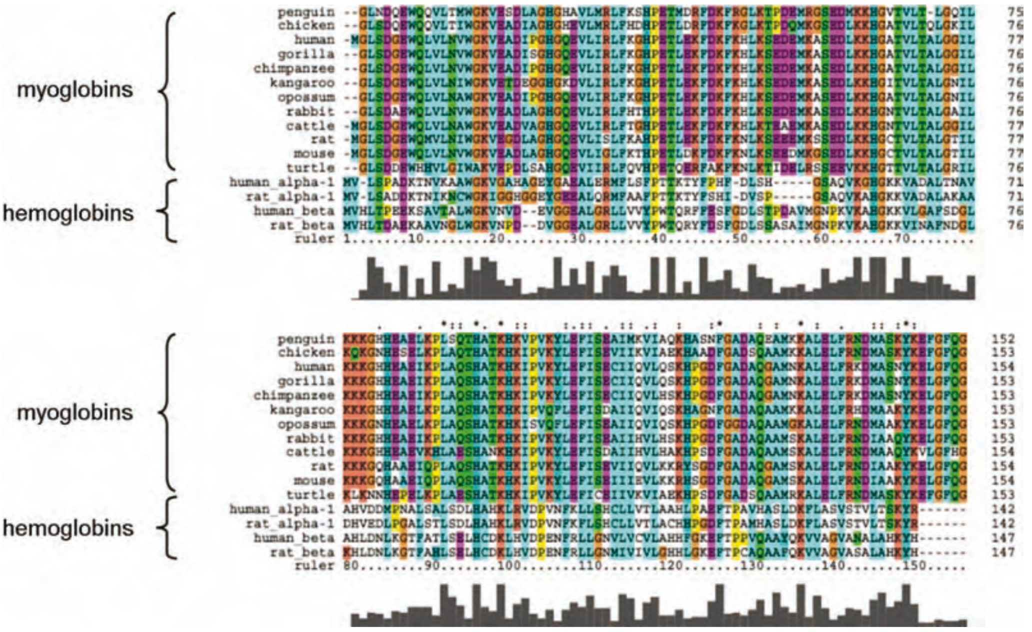
The question remains of how to determine whether approximation by an extreme-value distribution is admissible for a certain scoring scheme and gap penalty setting one is using. This can be tested on randomly generated (or, simply, unrelated) sequences by computing a global alignment between sequences under that particular parameter setting. If the result has a negative sign (averaged over many trials or on very long random sequences), then the approximation is admissible. This is based on a theorem due to Arratia and Waterman [6], and subsequent simulation results reported by Waterman and Vingron [84]. In particular, a gap open penalty of 12 with an extension penalty of 2 or 3 for the case of the PAM250 matrix, as well as any stronger combination, allows for approximation by the extreme-value distribution.

In database searching the fitting need not be done on randomly generated sequences. Under the assumption that the large majority of sequences in a database are not related to the query, the bulk of the scores generated upon searching can be used for fitting. This approach is taken by Pearson in the FASTA package. It has the advantage that the implicit random model is more realistic since it is taken directly from the data actually searched. Along a similar line of thought, Spang and Vingron [72] tested significance calculations in database searching by evaluating a large number of search results. Their study showed that one should not simply use the sum of the lengths of all the sequences in the database as the length parameter in the formula for the extreme-value distribution. This would overestimate the length that actually governs the statistics. Instead, a considerably shorter effective length can determined for a particular database using simulations. This effect is probably due to the fact that alignments cannot start in one sequence and end in the next one, which makes the number of feasible starting points for random alignments smaller than the actual length of the database.

5 Multiple Sequence Alignment

For many genes a database search reveals a whole number of homologous sequences. Then, one wishes to learn about the evolution and the sequence conservation in such a group. This question surpasses what can reasonably be achieved by the sequence comparison methods described in Section 3. Pairwise comparisons do not readily exhibit positions that are conserved among a whole set of sequences and tend to miss subtle similarities that become visible when observed simultaneously among many sequences. Thus, one wants to simultaneously compare several sequences.

A multiple alignment arranges a set of sequences in a scheme such that positions believed to be homologous are written in a common column (Figure 4). As in a pairwise alignment, when a sequence does not possess an amino acid in a particular position, this is denoted by a dash. There also are conventions similar to the ones for pairwise alignment regarding the scoring of a multiple alignment. The so-called sum-of-pairs (SOP) [2] score adds the scores of all the induced pairwise alignments contained in a multiple alignment. For a linear



```
      penguin --GLNDQEWQQVLTMWGKVESDLAGHGHAVLMRLFKSHPETMDRFDKFRGLKTPDEMRGSEDMKKHGVTVLT-LGQIL  75
      chicken --GLSDQEWQQVLTIWGKVEADIAGHGHEVLMRLFHDHPETLDRFDKFKGLKTPDQMKGSEDLKKHGATVLTQLGKIL  76
        human -MGLSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASEDLKKHGATVLTALGGIL  77
      gorilla --GLSDGEWQLVLNVWGKVEADISGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASEDLKKHGATVLTALGGIL  76
   chimpanzee --GLSDGEWQLVLNVWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASEDLKKHGATVLTALGGIL  76
     kangaroo --GLSDGEWQLVLNIWGKVETDEGGHGKDVLIRLFKGHPETLEKFDKFKHLKSEDEMKASEDLKKHGITVLTALGNIL  76
      opossum --GLSDGEWQLVLNAWGKVEADIPGHGQEVLIRLFKGHPETLEKFDKFKHLKSEDEMKASEDLKKHGATVLTALGNIL  76
       rabbit --GLSDAEWQLVLNVWGKVEADLAGHGQEVLIRLFHTHPETLEKFDKFKHLKSEDEMKASEDLKKHGNTVLTALGAIL  76
       cattle -MGLSDGEWQLVLNAWGKVEADVAGHGQEVLIRLFTGHPETLEKFDKFKHLKTEAEMKASEDLKKHGNTVLTALGGIL  77
          rat -MGLSDGEWQMVLNIWGKVEGDLAGHGQEVLISLFKAHPETLEKFDKFKNLKSEEEMKSSEDLKKHGCTVLTALGTIL  77
        mouse -MGLSDGEWQLVLNVWGKVEADLAGHGQEVLIGLFKTHPETLDKFDKFKNLKSEEDMKGSEDLKKHGCTVLTALGTIL  77
       turtle --GLSDDEWNHVLGIWAKVEPDLSAHGQEVIIRLFQVHPETQERFAKFKNLKTIDELRSSEEVKKHGTTVLTALGRIL  76
human_alpha-1 MV-LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF-DLSH-----GSAQVKGHGKKVADALTNAV  71
  rat_alpha-1 MV-LSADDKTNIKNCWGKIGGHGGEYGEEALQRMFAAFPTTKTYFSHI-DVSP-----GSAQVKAHGKKVADALAKAA  71
   human_beta MVHLTPEEKSAVTALWGKVNVD--EVGGEALGRLLVVYPWTQRFFESFGDLSTPDAVMGNPKVKAHGKKVLGAFSDGL  76
     rat_beta MVHLTDAEKAAVNGLWGKVNPD--DVGGEALGRLLVVYPWTQRYFDSFGDLSSASAIMGNPKVKAHGKKVINAFNDGL  76
        ruler 1.......10........20........30........40........50........60........70........

                  .    .  *:: *. * ::    :.:: . :: .:  :   :*    : :  * :  .   :: :*:
      penguin KKKGHHEAELKPLSQTHATKHKVPVKYLEFISEAIMKVIAQKHASNFGADAQEAMKKALELFRNDMASKYKEFGFQG  152
      chicken KQKGNHESELKPLAQTHATKHKIPVKYLEFISEVIIKVIAEKHAADFGADSQAAMKKALELFRNDMASKYKEFGFQG  153
        human KKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKHPGDFGADAQGAMNKALELFRKDMASNYKELGFQG  154
      gorilla KKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLQSKHPGDFGADAQGAMNKALELFRKDMASNYKELGFQG  153
   chimpanzee KKKGHHEAEIKPLAQSHATKHKIPVKYLEFISECIIQVLHSKHPGDFGADAQGAMNKALELFRKDMASNYKELGFQG  153
     kangaroo KKKGHHEAELKPLAQSHATKHKIPVQFLEFISDAIIQVIQSKHAGNFGADAQAAMKKALELFRHDMAAKYKEFGFQG  153
      opossum KKKGNHEAELKPLAQSHATKHKISVQFLEFISEAIIQVIQSKHPGDFGGDAQAAMGKALELFRNDMAAKYKELGFQG  153
       rabbit KKKGHHEAEIKPLAQSHATKHKIPVKYLEFISEAIIHVLHSKHPGDFGADAQAAMSKALELFRNDIAAQYKELGFQG  153
       cattle KKKGHHEAEVKHLAESHANKHKIPVKYLEFISDAIIHVLHAKHPSDFGADAQAAMSKALELFRNDMAAQYKVLGFHG  154
          rat KKKGQHAAEIQPLAQSHATKHKIPVKYLEFISEVIIQVLKKRYSGDFGADAQGAMSKALELFRNDIAAKYKELGFQG  154
        mouse KKKGQHAAEIQPLAQSHATKHKIPVKYLEFISEIIIEVLKKRHSGDFGADAQGAMSKALELFRNDIAAKYKELGFQG  154
       turtle KLKNNHEPELKPLAESHATKHKIPVKYLEFICEIIVKVIAEKHPSDFGADSQAAMRKALELFRNDMASKYKEFGFQG  153
human_alpha-1 AHVDDMPNALSALSDLHAHKLRVDPVNFKLLSHCLLVTLAAHLPAEFTPAVHASLDKFLASVSTVLTSKYR------  142
  rat_alpha-1 DHVEDLPGALSTLSDLHAHKLRVDPVNFKFLSHCLLVTLACHHPGDFTPAMHASLDKFLASVSTVLTSKYR------  142
   human_beta AHLDNLKGTFATLSELHCDKLHVDPENFRLLGNVLVCVLAHHFGKEFTPPVQAAYQKVVAGVANALAHKYH------  147
     rat_beta KHLDNLKGTFAHLSELHCDKLHVDPENFRLLGNMIVIVLGHHLGKEFTPCAQAAFQKVVAGVASALAHKYH------  147
        ruler 80........90.......100.......110.......120.......130.......140.......150.....
```

Figure 4 Example of a multiple sequence alignment: an alignment of amino acid sequences of myoglobins and hemoglobins from a number of species. Each sequence begins in the top block and continues in the bottom block. The color code indicates physicochemical attributes of amino acids. The bar diagram below the alignment quantifies the degree of conservation in the column above.

gap penalty this amounts to scoring each column of the alignment by the sum of the amino acid pair scores or gap penalties in this column. Although it would be biologically meaningful, the distinctions between global, local and other forms of alignment are rarely made in a multiple alignment. The reason for this will become apparent below when we describe the computational difficulties in computing multiple alignments.

In general, the columns of a multiple alignment cannot be determined based on the set of all pairwise alignments. Quite the contrary, pairwise alignments may contradict each other in that one set of alignments opts to place, say, residue *a* from sequence *i* in one column with residue *b* from sequence *j*, while from another set of pairwise alignments it may follow that *a* should be in one column with another letter *c* from sequence *j*. If one wishes to assemble a multiple alignment from pairwise alignments one has to avoid "closing loops", i.e. one can put together pairwise alignments as long as no new pairwise alignment is included involving a sequence which is already part of the multiple alignment. In particular, pairwise alignments can be merged when they align one sequence to all others, when a linear order of the given sequences is maintained or when the sequence pairs for which pairwise alignments are given form a tree. While all these schemes allow for the ready definition of algorithms that output multiply aligned sequences, they do not include any information stemming from the simultaneous analysis of several sequences.

An alternative approach is to generalize the dynamic programming optimization procedure applied for pairwise alignment to the delineation of a multiple alignment that maximizes, for example, the SOP score. The algorithm used [80] is a straightforward generalization of the global alignment algorithm. This is easy to see, in particular, for the case of the column-oriented SOP scoring function avoiding an affine gap penalty in favor of the simpler linear one. With this scoring, the arrangement of gaps and letters in a column can be represented by a Boolean vector indicating which sequences contain a gap in a particular column. Given the letters that are being compared, one needs to evaluate the scores for all these arrangements. However conceptually simple this algorithm may be, its computational complexity is rather forbidding. For *n* sequences it is proportional to 2^n times the product of the lengths of all sequences. The space requirement of this algorithm is on the order of the product over the lengths of the *n* sequences, which constitutes an even greater obstacle to its practical application.

There exists software to compare three sequences with this algorithm that additionally implements a space-saving technique [46]. For more than three sequences, algorithms have been developed that aim at reducing the search space while still optimizing the given scoring function. The most prominent program of this kind is MSA2 [25,44]. An alternative approach is used by DCA

[36, 73], which implements a "divide-and-conquer" philosophy. The search space is repeatedly subdivided by identifying anchor points through which the alignment is highly likely to pass.

However, none of these approaches scales well to large numbers of sequences to be aligned. The most common remedy is reducing the multiple alignment problem to an iterated application of the pairwise alignment algorithm. However, in doing so, one also aims at drawing on the increased amount of information contained in a set of sequences. Instead of simply merging pairwise alignments of sequences, the notion of a profile [24] has been introduced in order to grasp the conservation patterns within subgroups of sequences. A profile is essentially a representation of an already computed multiple alignment of a subgroup of sequences. This alignment is "frozen" for the remaining computation. Other sequences or other profiles can be compared to a given profile based on a generalized scoring scheme defined for this purpose. The advantage of scoring a sequence versus a profile over scoring individual sequences lies in the fact that the scoring schemes for profile matching reflect the conservation patterns among the already aligned sequences. (Profiles are discussed in more detail in Chapter 11.)

Given a profile and a single sequence, the two can be aligned using the basic dynamic programming algorithm together with the accompanying scoring scheme. The result will be an alignment between sequence and profile that can readily be converted into a multiple alignment now comprising the sequences underlying the profile plus the new one. Likewise, two profiles can be aligned with each other, resulting in a multiple alignment containing all sequences from both profiles. Various multiple alignment strategies can be implemented with these tools. Most commonly, a hierarchical tree is generated for the given sequences, which is then used as a guide for iterative profile construction and alignment. This alignment strategy is called "progressive", and was introduced in papers by Taylor [76], Corpet [16] and Higgins [28]. Higgins' program Clustal [42] and, in particular, its latest version ClustalW are probably the most widely used programs for multiple sequence alignment [47]. Two recent variants of progressive alignment are MUSCLE [21] and PROBCONS [19]. Other programs in practical use are the MSA2 program and DCA. Lee and coworkers [54] developed a program that focuses on fast alignment of highly similar sequences, e.g. ESTs, using an algorithm termed partial order alignment.

Progress has been made also on the problem of local multiple alignment. The algorithm behind the Dialign [37, 56] program relies on collecting local similarities among all pairs of sequences and then assembles those into multiply aligned regions. Similarly, T-Coffee [62] allows for inclusion of both local and global alignments, as well as other possible information like structural similarity, and merges those consistently into a multiple alignment.

Since iterative profile alignment tends to be guided by a hierarchical tree, this step of the computation also influences the final result. Usually the hierarchical tree is computed based on pairwise comparisons and their resulting alignment scores. Subsequently, this score matrix is used as input to a clustering procedure like single linkage clustering or UPGMA (unweighted pair group method with arithmetic mean) [74]. However, it is well understood that in an evolutionary sense such a hierarchical clustering does not necessarily result in a biologically valid tree. Thus, when allowing this tree to determine the multiple alignment there is the danger of pointing further evolutionary analysis of this alignment in the wrong direction. Consequently, the question has arisen of a common formulation of evolutionary reconstruction and multiple sequence alignment. The cleanest, although biologically somewhat simplistic, model attempts to reconstruct ancestral sequences to attribute to the inner nodes of a tree [65]. Such reconstructed sequences at the same time determine the multiple alignment among the sequences. In this "generalized tree alignment" one aims at minimizing the sum of the edge lengths of this tree, where the length of an edge is determined by the alignment distance between the sequences at its incident nodes. As to be expected, the computational complexity of this problem again makes its solution unpractical. The practical efforts in this direction go back to the work of Sankoff [65, 66]. Hein [26] and Schwikowski and Vingron [68] produced software [38, 40] relying on these ideas.

With the increased interest in analysis of regulatory regions in DNA, the problem of finding subtle local similarities, in particular in DNA sequences, has received much interest. Many programs for the detection of common sequence motifs use probabilistic modeling and/or machine learning approaches. In particular, the mathematical technique of the Gibbs sampler has lent its name also to a motif-finding program, the Gibbs Motif Sampler [31,53]. Bailey and Elkan [7] designed the MEME [33] program which relies on an expectation maximization algorithm. A number of pattern-finding programs have been compared by Tompa and coworkers [78].

6 Multiple Alignments, Hidden Markov Models (HMMs) and Database Searching II

Information about which residues are conserved and thus important for a particular family is crucial not only for the purpose of multiply aligning a set of sequences, but is also very valuable in the context of identifying related sequences in a database. A multitude of methods has been developed that aim at identifying sequences in a database which are related to a given family. The first one was the notion of a profile that was described above and was actually introduced in the context of database searching. As in multiple alignment, profiles help in emphasizing conserved regions in a database search. Thus, a sequence that matches the query profile in a conserved region will receive a higher score than a database sequence matching only in a divergent part of an alignment. This feature is of enormous help in distinguishing truly related sequences.

Algorithmically, profile searching simply uses the dynamic programming alignment algorithm for aligning a sequence to a profile on each sequence in the database. Of course, this is computationally quite demanding and much slower than the heuristic database search algorithms like BLAST or FASTA. Typically, the multiple alignment underlying the profile describes a conserved domain which one expects to find within a database sequence. Therefore, in this context, it is important that end gaps should not be penalized. Furthermore, gap penalties for profile matching frequently vary along the profile in order to reflect the existence of gaps within the underlying multiple alignment. Through this mechanism one attempts to allow new gaps preferentially in regions where gaps have been observed already. However, different suggestions exist as to the choice and derivation method for these gap penalties [77].

In 1994, Haussler and coworkers [52] and Baldi and coworkers [8] introduced HMMs for the purpose of identifying family members in a database. An HMM is a generative probabilistic model in the sense that we can think of it as a machine that generates strings of symbols; in biological applications, typically the letters of a biological sequence. It has "states" and each state will output a symbol according to a distribution associated to this state. After a state has output a symbol, a transition to one of its successor states occurs according to a specified transition probability. These transitions are Markovian, meaning that the transitions leading out of a state are governed only by this state's transition probabilities and not by how the machine got to arrive in this state. The "hidden" element in the HMM comes from the image that an observer gets to see the generated symbol series and then needs to infer which series of states gave rise to it or what the underlying distributions might look like. HMMs and related algorithms are discussed in depth by Durbin and coworkers [20].

The structure of a *profile HMM* mimics a multiple alignment. We think of it as a machine that emits a sequence which would typically be randomly drawn based on a given multiple alignment, according to the distribution of letters in its columns. If gaps were forbidden, the emitted sequence would essentially draw one letter from each column of the alignment. Insertions and deletions, however, imply that the generated sequence may differ in length from the multiple alignment, with some columns possibly skipped or new letters inserted in the emitted sequence. Figure 5 schematically shows the

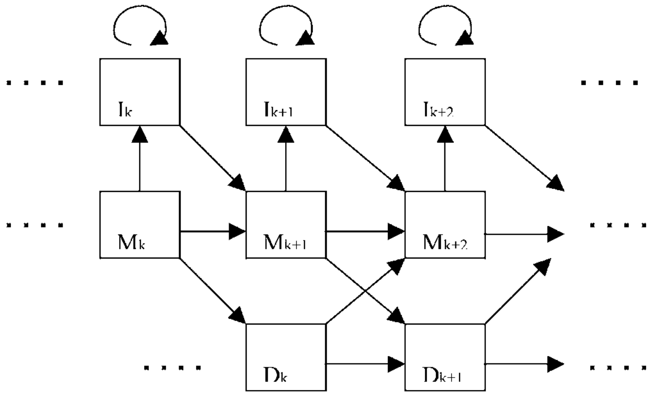


Figure 5 Sketch of the structure of a profile HMM.

states and transitions that realize this structure. The middle row represents a series of match states (M). These represent the columns of the given alignment and emit letters according to a distribution that is supposed to fit the corresponding column of the alignment. A transition into an insert state (denoted I, arranged in the top row) lets the machine emit an additional letter, with the possibility of remaining in this insert state and emitting more additional letters as indicated by the self-loop at the insert nodes. The transition from a match state into a delete state (D, bottom row) leads to the emitted sequence skipping one or more of the following columns of the alignment, which corresponds to a deletion in the emitted sequence with respect to the alignment.

In this manner, the profile HMM can output sequences which, by way of their generating state sequence, are aligned relative to the given multiple alignment. The task of aligning a sequence to a profile HMM can therefore be phrased in the probabilistic setting of "What is the most likely sequence of states to have given rise to this sequence?". This is solved by the so-called Viterbi algorithm, which largely resembles the classical dynamic programming sequence alignment algorithm in its structure. Alternatively, one can ask for the probability of the observed sequence as such, independent of which path generated it. This is computed by summing over the different state sequences that could have produced the sequence. Here, the fully probabilistic formalization is superior to an *ad hoc* score definition which would not allow for posing and answering this question. Algorithmically, this summation can be computed efficiently by the so-called forward algorithm.

There is a standard learning algorithm, the Baum–Welch algorithm, to determine emission and transition probabilities of an HMM given a set of learning data. When training a profile HMM, one has the sequences of the multiple alignment at hand, which may be too small a set for parameter determination in many cases. The problem becomes manageable, though, when one uses the residue distributions in the columns as a guideline for the emission probabilities and chooses the transition probabilities to reflect, in essence, the way gaps should be handled. Adding a possible correction (the "pseudocounts") for sampling artifacts, this choice of parameters can either be used directly or as a starting configuration for a subsequent application of the Baum–Welch algorithm in order to refine parameter estimation. Nevertheless, training an HMM is a very difficult problem and the Baum–Welch algorithm may only find a local optimum.

The first application of HMMs in sequence analysis seems to be due to Churchill [15], who applied the technique to the segmentation of sequences based on their composition. Profile HMMs followed later, addressing the same problem as multiple alignment profiles. A widely used implementation of profile HMMs is the HMMER package [32]. The two concepts of HMMs and profiles are formally very similar, although set in a different language. Bucher and Karplus [12] introduced generalized profiles, and showed that the two concepts are equally powerful in their abilities to model sequence families and detect related sequences. Nevertheless, due to the coherent probabilistic description language and a broad spectrum of good software implementations, HMMs have found widespread acceptance. Many other areas in computational molecular biology, e.g. gene finding, have also profited greatly from the introduction of HMMs.

The fact that a profile or HMM can pick out new sequences also related to the given family suggests that these should be used to update the profile or HMM used as search pattern. This idea leads to iterative search algorithms which search the database repeatedly, each time updating the query pattern with some or all of the newly identified sequences. PSI-BLAST [3] is a very successful implementation of this idea. It starts with a single sequence, and after the first search constructs profiles from conserved regions among the query and newly identified sequences. Without allowing for gaps (to increase search speed) these new profiles are used to repeat the search. Generally, PSI-BLAST quickly converges after updating these profiles again and generally is very successful in delineating all the conserved regions a sequence may share with other sequences in a database. In the realm of HMMs, SAM is a very careful implementation of the idea of iterated searches [39, 49].

It is the generally held view that searching a database with a profile or HMM produces extreme-value distributed random scores just like single-sequence database searching. The quality of the fit to the extreme-value distribution may, however, depend on the particular given alignment. This has been substantiated with mathematical arguments only for the case of ungapped profile matching [22]. Nevertheless, this basic understanding of the statistical behavior of database-matching methods is a crucial element of iterative search programs. Without clear and reliable cutoff values one could not decide which sequences to integrate into the next search pattern and would run the danger of including false positives, thus blurring the information in the pattern.

Both single-sequence search methods and profile/HMM-based methods have been thoroughly validated during recent years [11]. Databases of structurally derived families, e.g. SCOP [34, 58], have made it possible to search a sequence database with a query, and exactly determine the number of false positives and false negatives. For every search one determines how many sequences one misses (false negatives) in dependence of the number of false positive matches. If the sequence statistics is accurate, the number of false positives correlates well with the E-value, i.e. the number of false positives expected by chance. This way of validating search methods allows for making objective comparisons and for determining how much quality one actually gains with slower methods over faster, less accurate methods.

7 Protein Families and Protein Domains

The companion question to the one that assigns related sequences from a database to a given query sequence or family is the question that tries to assign to a query sequence the family of which it is a member or the domains that it contains. One resource for this purpose is the InterPro database [4], which contains amino acid patterns that are descriptive for particular domains, families or functions. The InterPro database summarizes information from several other motif databases including, among others, Prosite [30] and Pfam [10]. One can either scan a sequence against this database [86] or rely on precomputed information that is stored along with the sequences in the databases. The Pfam database contains precomputed HMMs for protein domains. A query sequence can be matched against this library of HMMs in order to identify known domains in the query sequence. Here, too, match statistics plays a crucial role in order to determine the significantly matching domains. A server that allows one to scan a sequence versus all Pfam domains can be found at the Sanger center [45]. Software has also been developed to recognize the Pfam HMMs in either coding DNA or in genomic DNA. In the latter case, the program combines the HMM matching with the distinction between coding and noncoding DNA.

Apart from finding and cataloguing domains of proteins, efforts have also been made to structure the space of all protein sequences into homologous groups or orthologous families. Linial and coworkers have developed the Protonet [67] system, hierarchically structuring the set of all proteins. Krause and coworkers [51] developed SYSTERS [35] to delineate protein families and supply consensus sequences of these families to be searched with a DNA or protein query sequence. Koonin and coworkers put special emphasis on the delineation of orthologous genes, and collect this information in the COG and KOG databases [75].

8 Conclusions

The problems and methods introduced above have been instrumental in the advance in our understanding of genome function, organization and structure. While some years ago human experts would check every program output, nowadays sequence analysis routines are being applied in an automatic fashion creating annotation that is included in various databases. This holds true for similarity relationships among sequences and extends all the way to the prediction of genomic structure or to function prediction based on similarity. Although the quality of the tools has increased dramatically, the possibility of error and, in particular, its perpetuation by further automatic methods exists. Thus, it is apparent that the availability of these high-throughput computational analysis tools is a blessing and a problem at the same time.

References

1 ALTSCHUL, S. F., W. GISH, W. MILLER, E. W. MYERS AND D. J. LIPMAN. 1990. Basic local alignment search tool. J. Mol. Biol. **215**, 403–10.

2 ALTSCHUL, S. F. AND D. J. LIPMAN. 1989. Trees, stars, and multiple biological sequence alignment. SIAM J. Appl. Math. **49**, 197–209.

3 ALTSCHUL, S. F., T. L. MADDEN, A. A. SCHAFFER, J. H. ZHANG, Z. ZHANG, W. MILLER AND D. J. LIPMAN. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**, 3389–402.

4 APPWEILER, R., T. K. ATTWOOD, A. BAIROCH, et al. 2001. The InterPro database, an integrated documentation resource for protein families, domains and functional sites. Nucleic Acids Res. **29**, 37–40.

5 ARGOS, P. 1987. A Sensitive procedure to compare amino-acid-sequences. J. Mol. Biol. **193**, 385–96.

6 ARRATIA, R. AND M. S. WATERMAN. 1994. A phase transition for the score in matching random sequences allowing deletions. Ann. Appl. Prob. **4**, 200–25.

7 BAILEY, T. L. AND C. ELKNA. 1995. The value of prior knowledge in discovering motifs with MEME. Proc. ISMB **3**, 21–9.

8 BALDI, P., Y. CHAUVIN, T. HUNKAPILLER AND M. A. MCCLURE. 1994. Hidden Markov-models of biological primary sequence information. Proc. Natl Acad. Sci. USA **91**, 1059–63.

9 BARTON, G. J. 1993. An efficient algorithm to locate all locally optimal alignments between two sequences allowing for gaps. Comput. Appl. Biosci. **9**, 729–34.

10 BATEMAN, A., E. BIRNEY, L. CERRUTI, et al. 2002. The Pfam protein families database. Nucleic Acids Res. **30**: 276–80.

11 BRENNER, S. E., C. CHOTHIA AND T. J. P. HUBBARD. 1998. Assessing sequence comparison methods with

reliable structurally identified distant evolutionary relationships. Proc. Natl Acad. Sci. USA **95**: 6073–6078.

12 BUCHER, P., K. KARPLUS, N. MOERI AND K. HOFMANN. 1996. A flexible motif search technique based on generalized profiles. Comput. Chem. **20**: 3–23.

13 BURKHARDT, S., A. CRAUSER, P. FERRAGINA, H.-P. LENHOF, E. RIVALS AND M. VINGRON. 1999. q-gram based database searching using a suffix array (QUASAR). Proceedings of the Third International Conference on Computational Molecular Biology (RECOMB), ACM Press, New York: 77–83.

14 CALIFANO, A. AND I. RIGOUTSOS. 1993. FLASH: a fast look-up algorithm for string homology. Proc. ISMB **1**: 56–64.

15 CHURCHILL, G. A. 1989. Stochastic-models for heterogeneous DNA-sequences. Bull. Math. Biol. **51**: 79–94.

16 CORPET, F. 1988. Multiple sequence alignment with hierarchical-clustering. Nucleic Acids Res. **16**: 10881–90.

17 DAYHOFF, M. O., W. C. BARKER AND L. T. HUNT. 1978. Establishing homologies in protein sequences. Atlas of Protein Sequences and Structure **5**: 345–52.

18 DEMBO, A., S. KARLIN AND O. ZEITOUNI. 1994. Critical phenomena for sequence matching with scoring. Ann. Prob. **22**: 2022–39.

19 DO, C. B., M. S. P. MAHABHASHYAM, M. BRUDNO AND S. BATZOGLOU. 2005. ProbCons: probabilistic consistency-based multiple sequence alignment. Genome Res. **15**: 330–40.

20 DURBIN, R., S. EDDY, A. KROGH AND G. MITCHISON. 1998. *Biological Sequence Analysis*. Cambridge University Press, Cambridge.

21 EDGAR, R. C. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. **32**: 1792–7.

22 GOLDSTEIN, L. AND M. S. WATERMAN. 1994. Approximations to profile score distributions. J. Comput. Biol. **1**: 93–104.

23 GOTOH, O. 1982. An improved algorithm for matching biological sequences. J. Mol. Biol. **162**: 705–8.

24 GRIBSKOV, M., A. D. MCLACHLAN AND D. EISENBERG. 1987. Profile analysis – detection of distantly related proteins. Proc. Natl Acad. Sci. USA **84**: 4355–8.

25 GUPTA, S. K., J. D. KECECIOGLY AND A. A. SCHAFFER. 1995. Improving the practical space and time efficiency of the shortest-path approach to sum-of-pairs multiple sequence alignment. J. Comp. Biol. **2**: 459–72.

26 HEIN, J. 1990. Unified approach to alignment and phylogenies. Methods Enzymol. **183**: 626–45.

27 HENIKOFF, S. AND J. G. HENIKOFF. 1992. Amino-acid substitution matrices from protein blocks. Proc. Natl Acad. Sci. USA **89**: 10915–9.

28 HIGGINS, D. G., A. J. BLEASBY AND R. FUCHS. 1992. Clustal-V – Improved software for multiple sequence alignment. Comput. Appl. Biosci. **8**: 189–91.

29 HIRSCHBERG, D. S. 1977. Algorithms for longest common subsequence problem. J. ACM **24**: 664–75.

30 HOFMANN, K., P. BUCHER, L. FALQUET AND A. BAIROCH. 1999. The PROSITE database, its status in 1999. Nucleic Acids Res. **27**: 215–9.

31 http://bayesweb.wadsworth.org/gibbs/gibbs.

32 http://hmmer.wustl.edu.

33 http://meme.sdsc.edu.

34 http://scop.mrc-lmb.cam.ac.uk/scop.

35 http://systers.molgen.mpg.de.

36 http://www.bibiserv.techfak.uni-bielefeld.de/dca.

37 http://www.bibiserv.techfak.uni-bielefeld.de/dialign.

38 http://www.bioweb.pasteur.fr/Seqanal/interfaces/treealign-simple.

39 http://www.cse.ucsc.edu/research/compbio/sam.

40 http://www.dkfz.de/tbi/services/3w/start.

41 http://www.ebi.ac.uk.

42 http://www.ebi.ac.uk/clustalw.

43 http://www.ncbi.nlm.nih.gov.

44 http://www.ncbi.nlm.nih.gov/CBBresearch/Schaffer/msa.

45 http://www.sanger.ac.uk.

46 Huang, X. Q. 1993. Alignment of three sequences in quadratic space. Appl. Comput. Rev. **1**: 7–11.

47 Jeanmougin, F., J. D. Thompson, M. Gouy and D. G. Higgins. 1998. Multiple sequence alignment with Clustal X. Trends Biochem. Sci. **23**: 403–5.

48 Karlin, S. and S. F. Altschul. 1990. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proc. Natl Acad. Sci. USA **87**: 2264–8.

49 Karplus, K., C. Barrett and R. Hughey. 1998. Hidden Markov models for detecting remote protein homologies. Bioinformatics **14**: 846–56.

50 Kent, W. J. 2002. BLAT – The BLAST-like alignment tool. Genome Res. **12**: 656–64.

51 Krause, A., J. Stoye and M. Vingron. 2005. Large scale hierarchical clustering of protein sequences. BMC Bioinformatics **6**: 15.

52 Krogh, A., M. Brown, I. S. Mian, Sjölander and D. Haussler. 1994. Hidden Markov models in computational biology – applications to protein modeling. J. Mol. Biol. **235**: 1501–31.

53 Lawrence, C. E., S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald and J. C. Wootton. 1993. Detecting subtle sequence signals – a Gibbs sampling strategy for multiple alignment. Science **262**: 208–14.

54 Lee, C., C. Grasso and M. F. Sharlow. 2002. Multiple sequence alignment using partial order graphs. Bioinformatics **18**: 452–64.

55 Maizel, J. V. and R. P. Lenk. 1981. Enhanced graphic matrix analysis of nucleic acid and protein sequences. Proc. Natl Acad. Sci. USA **78**: 7665–9.

56 Morgenstern, B. 1999. DIALIGN 2: improvement of the segment-to-segment approach to multiple sequence alignment. Bioinformatics **15**: 211–8.

57 Muller, T. and M. Vingron. 2000. Modeling amino acid replacement. J. Comput. Biol. **7**: 761–776.

58 Murzin, A. G., S. E. Brenner, T. Hubbard and C. Chothia. 1995. SCOP – a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. **247**: 536–40.

59 Myers, E. W. and W. Miller. 1988. Optimal alignments in linear-space. Comput. Appl. Biosci. **4**: 11–17.

60 Needleman, S. B. and C. D. Wunsch. 1970. A general method applicable to search for similarities in amino acid sequence of 2 proteins. J. Mol. Biol. **48**: 443–53.

61 Ning, Z., A. J. Cox and J. C. Mullikin. 2001. SSAHA: a fast search method for large DNA databases. Genome Res. **11**: 1725–9.

62 Notredame, C., D. G. Higgins and J. Heringa. 2000. T-Coffee: a novel method for fast and accurate multiple sequence alignment. J. Mol. Biol. **302**: 205–17.

63 Pearson, W. R. and D. J. Lipman. 1988. Improved tools for biological sequence comparison. Proc. Natl Acad. Sci. USA **85**: 2444–8.

64 Sankoff, D. 1972. Matching sequences under deletion/insertion constraints. Proc. Natl Acad. Sci. USA **69**: 4–6.

65 Sankoff, D. 1975. Minimal mutation trees of sequences. SIAM J. Appl. Math. **28**: 35–42.

66 Sankoff, D., R. J. Cedergren and G. Lapalme. 1976. Frequency of insertion–deletion, transversion, and transition in evolution of 5S ribosomal-RNA. J. Mol. Evol. **7**: 133–49.

67 Sasson, O., A. Vaaknin, H. Fleischer, E. Portugaly, Y. Bilu, N. Linial and M. Linial. 2003. ProtoNet: hierarchical classification of the protein space. Nucleic Acids Res. **31**: 348–52.

68 Schwikowski, B. and M. Vingron. 1997. The deferred path heuristic for the generalized tree alignment problem. J. Comput. Biol. **4**: 415–31.

69 Smith, T. F. and M. S. Waterman. 1981. Identification of common molecular subsequences. J. Mol. Biol. **147**: 195–7.

70 Smith, T. F., M. S. Waterman and C. Burks. 1985. The statistical distribution of nucleic acid similarities. Nucleic Acids Res. **13**: 645–56.

71 Sonnhammer, E. L. L. and R. Durbin. 1995. A dot-matrix program with dynamic threshold control suited for genomic DNA

and protein-sequence analysis. Gene-Combis **167**: 1–10.

72 SPANG, R. AND M. VINGRON. 1998. Statistics of large-scale sequence searching. Bioinformatics **14**: 279–84.

73 STOYE, J. 1998. Multiple sequence alignment with the divide-and-conquer method. Gene **211**: GC45–56.

74 SWOFFORD, D. L. AND G. J. OLSEN (eds.). 1990. *Phylogeny Reconstruction*. Sinauer Associates, Sunderland, MA.

75 TATUSOV, R. L., N. D. FEDOROVA, J. D. JACKSON, et al. 2003. The COG database: an updated version includes eukaryotes. BMC Bioinformatics **4**: 41.

76 TAYLOR, W. R. 1987. Multiple sequence alignment by a pairwise algorithm. Comp. Appl. Biosci. **3**: 81–7.

77 TAYLOR, W. R. 1996. A non-local gap-penalty for profile alignment. Bull. Math. Biol. **58**: 1–18.

78 TOMPA, M., N. LI, T. L. BAILEY, et al. 2005. Assessing computational tools for the discovery of transcription factor binding sites. Nat. Biotechnol. **23**: 137–44.

79 VINGRON, M. AND M. S. WATERMAN. 1994. Sequence alignment and penalty choice – review of concepts, case-studies and implications. J. Mol. Biol. **235**: 1–12.

80 WATERMAN, M. S. 1984. Efficient sequence alignment algorithms. J. Theor. Biol. **108**: 333–7.

81 WATERMAN, M. S. 1995. *Introduction to Computational Molecular Biology*. Chapman & Hall, London.

82 WATERMAN, M. S. AND M. EGGERT. 1987. A new algorithm for best subsequence alignments with application to transfer RNA–ribosomal-RNA comparisons. J. Mol. Biol. **197**: 723–8.

83 WATERMAN, M. S. AND M. VINGRON. 1994. Rapid and accurate estimates of statistical significance for sequence database searches. Proc. Natl Acad. Sci. USA **91**: 4625–8.

84 WATERMAN, M. S. AND M. VINGRON. 1994. Sequence comparison significance and Poisson approximation. Stat. Sci. **9**: 367–81.

85 WOZNIAK, A. 1997. Using video-oriented instructions to speed up sequence comparison. Comput. Appl. Biosci. **13**: 145–50.

86 ZDOBNOV, E. M. AND R. APWEILER. 2001. InterProScan – an integration platform for the signature-recognition methods in InterPro. Bioinformatics **17**: 847–8.

4
Phylogeny Reconstruction

Ingo Ebersberger, Arndt von Haeseler, and Heiko A. Schmidt

1 Introduction

In 1973 Theodosius Dobzhansky said "Nothing in biology makes sense except in the light of evolution" [27]. Although more than 30 years old, the citation still remains valid. Biologists nowadays use the massive amount of sequence data to infer the phylogenetic relationship of contemporary organisms. DNA, long words over a finite alphabet of four nucleotides, is transmitted from one generation to the next. The copying process and environmental factors lead to an accumulation of mutations in the sequence. Such mutations manifest themselves as slight changes in the DNA sequence, so-called *substitutions*. The vertical transmission (in time) of DNA together with the discrete nature of the mutations makes the molecule an ideal target to study phylogenetic relationship of organisms. Consequently, sequence-based phylogenies of organisms have been determined from many different genes. Such gene-trees have provided surprisingly new and sometimes controversial insights into the evolutionary relationships of organisms. However, research and debates still focus on the best methodology. That is, how do we measure similarity or dissimilarity, how can we model the process of substitution, how can we accurately infer the tree? Despite this ongoing discussion, molecular phylogenies are nowadays a routine tool for biologists interested in the evolution of organisms.

Moreover, the application of molecular phylogenies goes beyond the reconstruction of phylogenetic trees for organisms. Gene trees or, more general, sequence trees serve as an important source of information to understand how sequences are related. From this relatedness it is then possible to infer the function of an unknown sequence (see also Chapter 32). Not only function can be inferred, but also structure can be deduced from trees (Chapter 11). From sequence trees we can deduce the evolutionary history of the sequences themselves. We can determine regions that are conserved or highly variable and we can detect sequences that show a highly aberrant substitution pattern. Moreover, we can detect duplications of genes or parts of the genome; thus,

Bioinformatics - From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

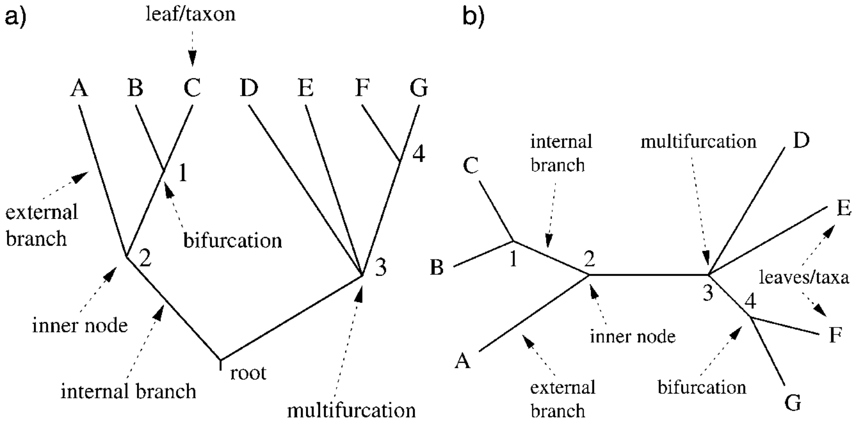


Figure 1 Phylogenetic relationships among a set of seven taxa represented by a rooted (a) and an unrooted (b) tree.

trees serve as analytical tools in comparative genomics (Chapter 37). Even the coevolution of host–pathogen interactions (Chapter 41) is mirrored in the similarity of trees from both groups.

Again, the conclusions strongly depend on the accuracy of the reconstruction method. To critically evaluate the result a basic understanding of the different approaches to phylogenetic inference is required. In this chapter we present a very basic introduction. We set the stage by a brief introduction in the terminology, followed by a summary about current approaches to model evolutionary changes (Section 2). Section 3 describes the three fundamental principles of phylogenetic inference, i.e. maximum parsimony, distance-based and maximum likelihood (ML) inference. The subsequent section deals with the optimization problem of finding the "best" tree(s) with respect to some objective function. With the advent of phylogenomics one wants to reconstruct a species tree from a collection of multiple genetic loci. This question leads naturally to supertree methods introduced in Section 5. Finally, Section 6 summarizes attempts to infer evolutionary relationships if the data do not evolve according to a tree. Processes like horizontal gene transfer or recombination destroy the tree-likeness of the data. In such instances it is better to reconstruct networks rather then trees.

1.1 Reconstructing a Tree from its Leaves

The fundamental axiom in evolutionary biology is the assumption that any two taxa share a common ancestor at some time point in their history. Thus, following backward in time the lineages along which these taxa have evolved, they will eventually coalesce. Considering a large set of taxa, consecutive coalescent events result in an ever-decreasing number of predecessors until only two lineages remain. The ancient taxon in which these lineages eventually merge then represents the most recent ancestor common to all taxa in the dataset. The correspondence to a tree is apparent and it is therefore of little surprise that trees play the key role in phylogenetic research. The dominant role of trees in the area of phylogeny reconstruction is already manifested in the phylogeneticist's vocabulary (Figure 1). Contemporary taxa are dubbed *leaves*, leaves are connected via *external branches* to *internal nodes* (their common ancestors) and internal nodes themselves are connected via *internal branches*. Nodes that give rise to two descendants are termed *bifurcations*, nodes with a larger number of descendants are referred to as *multifurcations* (Biologists usually judge it as unlikely that three or more lineages emerged at precisely the same time from a shared ancestor; thus, we will concentrate on bifurcating trees.) Eventually, if the direction of the evolutionary process is known the ancestor of all leaves in the tree is identified. To stay in the picture, this node is termed the *root*. If directional information is not available, the relationships of the taxa is represented in an *unrooted tree*. However, in this case the temporal succession of ancestors remains undetermined. The reconstruction of the phylogenetic relationships for a set of taxa and their representation by a tree can be separated into two subproblems. (i) What is the order individual taxa split from their shared ancestors, i.e. what is the topology of the tree? (ii) What is the evolutionary time that has passed since any two taxa last shared a common ancestor, i.e. how long are the corresponding branches of the tree? In most cases no hard evidence (such as a comprehensive fossil record) exists to directly reconstruct the evolutionary steps transforming one ancestral taxon into its descendants. Rather, we get hold only of the end-points of this process and are more or less ignorant about anything that has happened in the past. Thus, we are facing the problem of reconstructing the phylogenetic tree just by looking at its leaves.

1.2 Phylogenetic Relationships of Taxa and their Characters

Although one is typically interested in the relationships of the taxa, the reconstruction procedure is usually not based on the taxon as a whole. For practical reasons one vicariously concentrates on individual characteristics of these taxa, usually either morphological or molecular features. We will refer to such representative characteristics as *characters* and to their peculiar expression in the individual taxa as the *character state*.

To collect the raw data for phylogenetic analyses, the variety of states for a particular character in a set of taxa has to be assessed first. Next, the possible transformations of the character states during evolution has to be reconstructed, which can then be used for phylogenetic inference. Irrespective of what type of character has been chosen, two general approaches can be followed to trace the phylogenetic signal in the data. One can identify those (evolutionary novel) character states that are shared among a subset of the taxa. Such a congruency is interpreted as a result of shared descent [68]. Based on the pattern and extent of congruent character states, the degrees of relationships among the taxa can then be inferred. Alternatively, the extent of evolutionary change for the particular character between any pair of taxa can be assessed. From the resulting pairwise evolutionary distances again a phylogenetic tree can be reconstructed. We will outline both approaches in greater detail below (Section 3). Eventually, the evolutionary history of the particular character is extrapolated to the taxon level.

Inherent to any character-based strategy for phylogeny reconstruction is the assumption that comparisons are performed only between homologous characters, i.e. characters related by descent. Although the assignment of homology constitutes one of the major issues in evolutionary studies [68, 114, 134], we will take for granted that this postulation is met.

1.2.1 The Problem of Character Inconsistencies

To date, tree reconstruction is frequently based on more than only a single character. This adds the advantage that different characters can complement each other by adding resolution to different parts of the tree. However, the reverse effect occurs as well: trees reconstructed from different characters can disagree. Given that incompatible groupings of taxa are supported significantly by the respective data, two alternative explanations are possible. First, the incompatible groupings are based on a misinterpretation of the data. For example, taxa can share the same character state not due to a shared ancestry, but rather because the particular state arose independently at least twice during evolution. Phylogeny reconstruction methods that model the evolutionary process (see Sections 3.2 and 3.3) usually account for this problem. However, if such parallel, convergent or back mutations remain unrecognized (or are neglected) an erroneous tree reconstruction is possible. In the second explanation, the evolution of each character state is correctly reconstructed. Such genuine discrepancies between inferred trees have various causes. Among the most frequently stated are processes like the random sorting of ancestral polymorphisms (e.g. Ref. [130]) and horizontal gene transfer [25, 28, 91] (Figure 2). If one is suspicious that either of these scenarios could apply, several independently evolving characters should be analyzed. The most frequently observed tree is then usually the tree reflecting the evolutionary relationships of the taxa as a whole. Alternatively, if it seems appropriate to visualize such discrepant phylogenetic signals in the data, a network rather than a tree can be chosen to represent the phylogeny of the taxa (see Section 6).

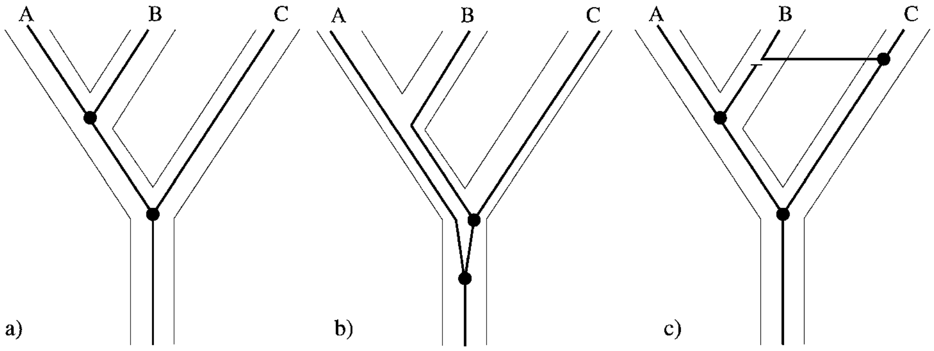


Figure 2 Trees for individual characters (inner trees) can differ from the species tree (outer trees). (a) The phylogenetic history of the character follows that of the species. (b) The random sorting of ancestral polymorphisms at subsequent speciation events. (c) Horizontal gene transfer, i.e. the lateral transfer of individual genes or DNA sequences between species. Both random sorting of ancestral polymorphisms and horizontal gene transfer can result in phylogenies that are incongruent to the species tree or to trees reconstructed from other characters.

1.2.2 **Finding the Appropriate Character Set**

In theory, phylogenetic relationships can be reconstructed from any set of homologous characters subject to evolutionary change. Depending on the scope of a study and the collection of taxa, however, certain types of characters might be more suitable than others. Changes in shape, or morphology in general, are the most conspicuous effects of evolution. Therefore, the field of phylogeny reconstruction was dominated by the analysis of morphological characters for a long time. However, with the expansion of molecular biology the focus has shifted considerably. Initially, the immunological and electrophoretic analysis of structural and electrical properties of proteins [85], presence or absence of genomic features such as restriction enzyme recognition sites [47], or DNA–DNA hybridization [45, 137] were used to measure the extent of character change on the molecular level. As time proceeded, the presence and absence or linear order of regulatory elements and genes (see also Chapter 8), and recently even expression data [83] have been employed for phylogenetic inference. However, the most dominant role is still taken by the direct comparison of biological sequences. Sequences change in the course of time and any two sequences derived from a common ancestor will diverge. The pattern and extent of differences between two related sequences is then used to reconstruct their evolutionary history. Initially, due to experimental constraints, the comparison of protein sequences was prevalent. Nowadays, analyses rely almost entirely on DNA sequences and even those studies comparing amino acid sequences usually derive these from the corresponding DNA sequences. The advantages of DNA sequence data are apparent. DNA sequences can be obtained with considerable ease from any taxon and even the comparison of entire genomes has now become feasible. Allowing for some simplifications, each nucleotide position in the DNA sequence can be regarded as an independently evolving character and the number of possible states is strictly limited to the four bases: adenine (A), guanine (G), cytosine (C) and thymine (T). Eventually, different DNA sequences in a genome can evolve with different rates. This allows for easy adaption of the dataset to the evolutionary time scale for which phylogenetic resolution is required.

2 Modeling DNA Sequence Evolution

The substitution of nucleotides in a DNA sequence, i.e. the replacement of one nucleotide by a different one, is usually considered a random event. As a consequence, an important prerequisite for the reconstruction of phylogenetic relationships among species is the prior specification of a *model of substitution*, which provides a statistical description of DNA sequence evolution [97]. If we consider the substitution of one nucleotide by another one at any given site in a sequence as a random event and, furthermore, assume that a series of such random events occurs during some time interval, then theses events form a homogeneous Poisson process [37], if two very mild assumptions are met:

(i) The occurrence of a substitution in the time interval (t_1, t_2) is independent of a substitution in another time interval (t_3, t_4), where (t_1, t_2) and (t_3, t_4) do not overlap.

(ii) There is a constant $\mu > 0$, such that for any time interval $(t, t + h)$, $h > 0$ and h small, the probability that one event occurs is independent of t and is proportional to μh. The probability that more than one substitution occurs during $(t, t + h)$ becomes vanishing small as $h \to 0$.

The latter condition implies the so-called time homogeneity and, moreover, it implies that the probability of one substitution is proportional to the length of the time interval, i.e. the size of h. As substitutions are assumed to occur spontaneously and independently from past or future substitutions, homogenous Poisson processes are a simple approach to model the evolution of DNA. Moreover, under conditions (1) and (2) the number of substitutions $X(t)$ that occur up to any arbitrary time t is Poisson distributed with parameter μt [37]. Thus:

$$P_i(t) = [(\mu t)^i \exp(-\mu t)]/i! \,, \tag{1}$$

is the probability that $i = 0, 1, 2, \ldots$ substitutions occur in the time interval $(0, t)$. On average, μt substitutions with variance μt are expected. Note that the parameters μ (nucleotide substitutions per site per unit time) and t (the time) cannot be estimated separately, but only through their product μt (number of substitutions per site up to time t).

The nucleotide substitution process of DNA sequences described by the Poisson process can be generalized to a so-called Markov process that uses a rate matrix (typically called Q with elements Q_{xy}), which specifies the relative rates of change for each nucleotide. The most general form of the Q-matrix is shown in Figure 3. Rows follow the order A, C, G and T so that, for example, the second term of the first row is the instantaneous rate of change from base A to base C. This rate is given by the frequency of base C (π_C) times a relative rate parameter, describing (in this case) how often the substitution A to C occurs during evolution with respect to all other possible substitutions. Thus, each nondiagonal entry in the matrix represents the flow from nucleotide x to nucleotide y, while the diagonal elements are chosen to make the sum of each row equal to zero. They represent the total flow that leaves nucleotide x. Accordingly, we can write the total number of substitutions per unit time (i.e. the total substitution rate μ) as:

$$\mu = -\sum_x Q_{xx}\pi_x, x \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}. \tag{2}$$

Models like the one summarized in Figure 3 belong to the general class of time-homogenous time-continuous Markov models. When applied to modeling nucleotide substitutions, they share the following set of assumptions:

- The rate of change from x to y at any nucleotide position in a sequence is independent of the nucleotide that occupied this position prior to x (Markov property).
- Substitution rates do not change over time (homogeneity).
- The waiting time until the first substitution occurs follows a continuous distribution (time continuity).

$$Q = \begin{pmatrix} -(a\pi_{\mathrm{C}} + b\pi_{\mathrm{G}} + c\pi_{\mathrm{T}}) & a\pi_{\mathrm{C}} & b\pi_{\mathrm{G}} & c\pi_{\mathrm{T}} \\ g\pi_{\mathrm{A}} & -(g\pi_{\mathrm{A}} + d\pi_{\mathrm{G}} + e\pi_{\mathrm{T}}) & d\pi_{\mathrm{G}} & e\pi_{\mathrm{T}} \\ h\pi_{\mathrm{A}} & i\pi_{\mathrm{C}} & -(h\pi_{\mathrm{A}} + i\pi_{\mathrm{C}} + f\pi_{\mathrm{T}}) & f\pi_{\mathrm{T}} \\ j\pi_{\mathrm{A}} & k\pi_{\mathrm{C}} & l\pi_{\mathrm{G}} & -(j\pi_{\mathrm{A}} + k\pi_{\mathrm{C}} + l\pi_{\mathrm{G}}) \end{pmatrix}$$

Figure 3 Instantaneous rate matrix Q. Each entry in the matrix represents the instantaneous substitution rate from nucleotide x to nucleotide y (rows and columns follow the order A, C, G and T). a to l are rate parameters describing the relative rate one nucleotide is substituted by any other nucleotide. $\pi_{\mathrm{A}}, \pi_{\mathrm{C}}, \pi_{\mathrm{G}}, \pi_{\mathrm{T}}$ correspond to the nucleotide frequencies. Diagonal elements are chosen such that each row sums up to zero.

Once the evolutionary model (and thus the Q-matrix) is specified, the probabilities $P(t)$ of change from one nucleotide to any other during evolutionary time t is computed as follows:

$$P(t) = \exp(Qt) \ . \tag{3}$$

Each entry $P_{xy}(t)$ of the resulting probability matrix $P(t)$ specifies the probability to observe nucleotide y at time point t if the original nucleotide at this site was x.

2.1 Nucleotide Substitution Models

From the instantaneous substitution rate matrix Q in Figure 3 various submodels can be derived. Among these, the so-called stationary time-reversible models are the ones most commonly used. These models introduce the constraint that for any two nucleotides i and j the rate of change from i to j is the same as from j to i (thus, $a = g, b = h, c = j, d = i, e = l, f = l$ in Figure 3). Under these conditions the values of π_N ($N = \mathrm{A, G, C, T}$) correspond to the stationary frequencies of the four nucleotides, respectively (i.e. $\pi \cdot Q = 0$). If all eight parameters of a reversible Q-matrix are specified separately, the general time reversible model [92] is derived. The most simplest (fewest number of parameter) model assumes that the equilibrium frequencies of the four nucleotides are 0.25 each and that any nucleotide has the same rate to be replaced by any other. These assumptions correspond to a Q-matrix with $\pi_\mathrm{A} = \pi_\mathrm{C} = \pi_\mathrm{G} = \pi_\mathrm{T} = 1/4$, and $a = b = c = d = e = f = 1$. This resembles the well-known Jukes-Cantor model [82]. An overview of the hierarchy of the most common substitution models is shown in Figure 4.

2.2 Modeling Rate Heterogeneity

The nucleotide substitution models described so far implicitly assume that the rate of nucleotide substitution is the same for any position in the DNA sequence. However, it is well known that this is an oversimplification. For example, substitutions occur at an about 10 times higher frequency at C and G nucleotides when the C is followed by a G along the sequence [71]. Similarly, selective constraints maintaining functional DNA sequences result in varying substitution rates along a DNA sequence. To account for such site-dependent rate variations, a plausible model for the distribution of rates over sites is required. Most commonly, a continuous probability distribution, the Γ-distribution with expectation 1 and variance $1/\alpha$, is used [61]. By adjusting the shape parameter α, the Γ-distribution allows varying degrees of rate heterogeneity (Figure 5). For $\alpha > 1$, the distribution is bell-shaped and models weak rate heterogeneity among sites. For $\alpha < 1$, the Γ-distribution takes on

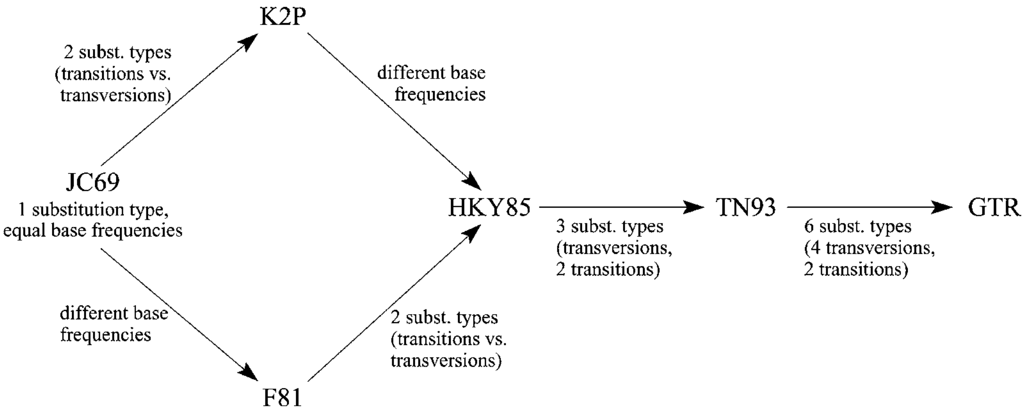

K2P
2 subst. types (transitions vs. transversions)
different base frequencies
JC69
1 substitution type, equal base frequencies
HKY85
3 subst. types (transversions, 2 transitions)
TN93
6 subst. types (4 transversions, 2 transitions)
GTR
different base frequencies
2 subst. types (transitions vs. transversions)
F81

Figure 4 A hierarchy of the most commonly used nucleotide substitution models. JC69: Jukes and Cantor (1969) [82]; F81: Felsenstein (1981) [42]; K2P: Kimura two-parameter model (1980) [84]; HKY85: Hasegawa, Kishino and Yano (1985) [65]; TN93: Tamura and Nei (1993) [149]; GTR: general time reversible model [92]. Many more models are possible and an extensive overview is given in Ref. [74].

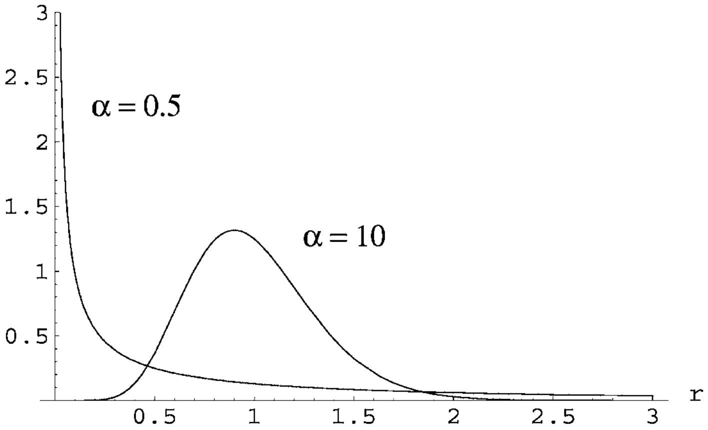


Figure 5 Probability density functions of the Γ-distribution for different values of the shape parameter α. The x-axis represents the relative substitution rate r of a site.

a characteristic L-shape, which describes strong rate heterogeneity, i.e. some sites have very high substitution rates, while the other sites are practically invariable.

2.3 Codon Models

Heterogeneous substitution rates become a particular issue for DNA that codes for proteins. Amino acid sites in a protein sequence are under different selective constraints, depending on their relevance for the protein func-

$$q_{xy} = \begin{cases} 0 & \text{if } x \text{ and } y \text{ differ at two or three nucleotide positions} \\ \pi_y & \text{if } x \text{ and } y \text{ differ by one synonymous transversion} \\ \kappa\pi_y & \text{if } x \text{ and } y \text{ differ by one synonymous transition} \\ \omega^{(h)}\pi_y & \text{if } x \text{ and } y \text{ differ by one nonsynonymous transversion} \\ \omega^{(h)}\kappa\pi_y & \text{if } x \text{ and } y \text{ differ by one nonsynonymous transition} \end{cases}$$

Figure 6 Instantaneous rate that codon x at site h is replaced by codon y. π_y represents the stationary marginal frequency of codon y, κ denotes the transition/transversion rate ratio and $\omega^{(h)}$ the nonsynonymous/synonymous substitution rate ratio at site h.

tion. Accordingly, nucleotide substitutions causing the encoded amino acid to change (replacement substitutions) will have fixation probabilities that depend on the selective constraint imposed on the encoded amino acid. In contrast, silent substitutions, i.e. a change in the DNA sequence has no effect on the encoded protein, are invisible to selective forces acting on the protein sequence. As a result, the ratio of nonsynonymous and synonymous substitution rates (ω) will vary among sites in a DNA sequence, with $\omega = 1$ indicating no selection, $\omega < 1$ representing purifying selection by removing replacement mutations and $\omega > 1$ representing diversifying positive selection/adaptive evolution. Codon models have been specifically designed to model the evolution of protein-coding DNA sequences [59]. An example is shown in Figure 6 based on an extension of the HKY85 model [65] (see Figure 4). Note that, in contrast to the conventional substitution models, codon models consider the replacement of one nucleotide triplet (codon) by another. Thus, we obtain $4^3 - 3 = 61$ possible character states at a site, the codon (the three stop codons are not taken into account). Obviously, the assignment of a distinct substitution rate ratio ω to each codon position would lead to a vast overparameterization of the model. Therefore, either a set of predefined ω-values or statistical distributions, both discrete and continuous, are used to account for varying ω-values among sites [104,156,157].

3 Tracing the Evolutionary Signal

Given a set of homologous DNA sequences whose phylogeny is known, inferences can be made about the evolutionary forces molding the contemporary DNA sequences from their shared ancestral sequence. Conversely, with a concept or a model at hand of how DNA sequences evolve one can aim to reconstruct the phylogenetic tree based on the DNA sequences. In either case, however, a meaningful sequence alignment is required. Thus, the sequences need to be aligned such that homologous nucleotides in different sequences form a column. To account for the insertion and deletion of nucleotides during evolution, gaps are introduced to achieve this positional homology. Chapter

3 deals with methods to compute sequence alignments. We will not dwell on further methodological details and take it for granted that an alignment is available. Based on this alignment, several criteria exist to compute the tree that best reflects the evolutionary relationships of the data. We will explain four principles for tree generation.

3.1 The Parsimony Principle of Evolution

Parsimony methods share as an optimality criterion that among various alternative hypotheses the one that requires the minimal number of assumptions should be chosen. In the context of DNA sequence evolution one searches for the tree(s) that explain the observed diversity in the contemporary sequences with the minimal number of nucleotide substitutions (Figure 7). Usually only a fraction of the differences in a sequence alignment determine the so-called parsimonious tree(s). They are called phylogenetically informative in a parsimony analysis. For instance, position 8 in the alignment (Figure 7) best supports a tree grouping sequences *W* and *X*, and *Y* and *Z*, respectively. In

	Position								
	1	2	3	4	5	6	7	8	9
Sequence W:	C	G	C	A	C	T	G	T	T
Sequence X:	C	G	C	A	C	T	G	T	T
Sequence Y:	T	G	A	A	C	T	G	C	T
Sequence Z:	C	G	G	A	C	T	G	C	T
	*		*					*	

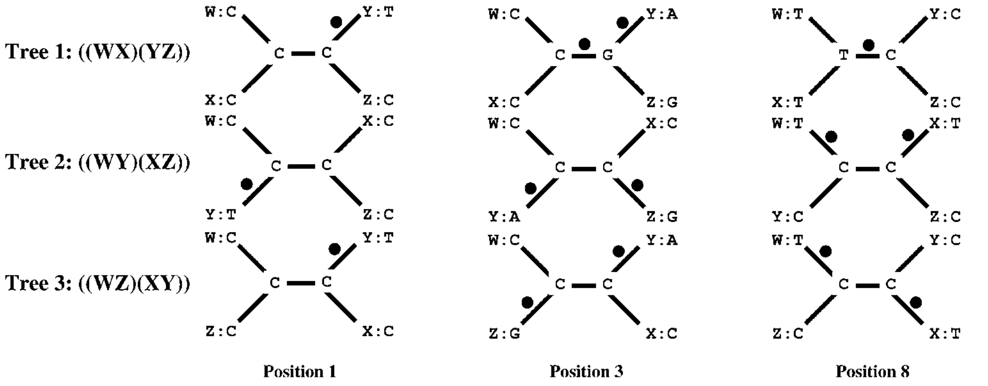


Figure 7 Maximum parsimony tree reconstruction from an alignment of four DNA sequences. For the alignment columns labeled with a "*" all three possible unrooted tree topologies are shown. Labels at the leafs denote the taxon and the represented nucleotide. Nucleotides at the inner nodes represent one parsimonious reconstruction. Nucleotide substitutions are represented by black dots. Position 8 is the only position that distinguishes the three trees with respect to the number of substitutions. Tree 1 requires only a single substitution compared to trees 2 and 3, which require two substitutions each.

contrast, to explain the sequence diversity at positions 1 and 3, two substitutions are necessary regardless of the tree structure. Such positions are called phylogenetically non-informative in a parsimony analysis. The tree that is supported best by the phylogenetically informative sites is then the maximum parsimony tree. Note, however, that no unique solution is guaranteed since more than one most parsimonious tree might exist.

3.1.1 Generalized Parsimony

To date, a vast number of modifications of the initial criterion of maximum parsimony exist [39,40,50,88]. Instead of referring to each and every modification separately, we would like to present the generalized idea of parsimony [128,129] from which the individual modifications can be easily derived. In a mathematical terminology, one aims to identify those trees in the space of all possible trees which minimize the following equation:

$$L(\tau) = \sum_{k=1}^{B} \sum_{j=1}^{L} \omega_j \cdot \mathrm{diff}(x_{k'j}, x_{k''j}) \ , \tag{4}$$

where $L(\tau)$ is called the length of the tree τ, B is the total number of branches in the tree, L is the number of nucleotide positions analyzed (alignment length), and k' and k'' are the two nodes connected by branch k displaying the nucleotides $x_{k'j}$ and $x_{k''j}$. These can be either the observed nucleotides present in the alignment or, in the case of internal nodes, the optimal nucleotide assignments. Finally, $\mathrm{diff}(x, y)$ represents the cost-matrix that specifies the cost of the transformation from nucleotide x to nucleotide y and ω_j is a specific weight for each alignment position. Thus, diff and $\omega = (\omega_1, \ldots, \omega_L)$ allow for specifying *a priori* some beliefs about the importance of positions and substitutions for the tree reconstruction, e.g. cost matrix **A** in Figure 8 reflects a Jukes–Cantor type of evolution, whereas cost matrix **B** down-weights transitions relative to transversions.

$$\mathbf{A} = \begin{array}{c|cccc} & A & C & G & T \\ \hline A & - & 1 & 1 & 1 \\ C & 1 & - & 1 & 1 \\ G & 1 & 1 & - & 1 \\ T & 1 & 1 & 1 & - \end{array} \qquad \mathbf{B} = \begin{array}{c|cccc} & A & C & G & T \\ \hline A & - & 5 & 1 & 5 \\ C & 5 & - & 5 & 1 \\ G & 1 & 5 & - & 5 \\ T & 5 & 1 & 5 & - \end{array}$$

Figure 8 Cost matrices for generalized parsimony. In matrix **A** substitutions between all four nucleotides invoke the same cost. Matrix **B** represents a slightly more sophisticated model. More weight is put on transversions than on transitions.

3.1.2 Multiple/Parallel Hits

Parsimony principles rely on the assumption that a group of related sequences share a certain nucleotide by descent. However, this only approximates the true evolutionary events if the overall amount of sequence changes is low. Thus, multiple changes at the same site in one taxon or parallel independent changes at the same site in different taxa are sufficiently infrequent not to be an issue. However, when considerably diverged sequences are used for tree reconstruction or marked substitution rate heterogeneity among sites exists, multiple/parallel hits can cause severe problems in both assessing the correct number of nucleotide substitutions along the phylogenetic tree and in inferring the correct tree topology.

3.2 Distance-based Methods

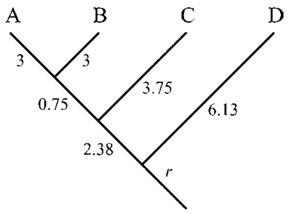
In contrast to parsimony methods with a biologically motivated approach to tree reconstruction, distance-based methods choose a mathematical route [43]. A phylogenetic tree is reconstructed for a set of taxa from their pairwise evolutionary distances. To this end, a distance matrix D is calculated from all possible pairwise sequence comparisons. Entry D_{ij} of this matrix represents the distance between sequence i and sequence j. In a simple approach D_{ij} is computed as the edit distance (Hamming distance), i.e. the minimum number of substitutions required to transform sequence i into j. However, multiple changes at the same position cannot be accounted for and therefore the Hamming distance will sometimes underestimate the true number of substitutions. To rectify this, models of sequence evolution are invoked that correct for multiple changes (see Section 2). Various methods were suggested for inferring a tree from a distance matrix. Common, although in fact not especially designed for phylogenetic tree reconstruction, are *clustering methods*. Clustering methods do not have an explicit objective function to be optimized. UPGMA, the most widespread clustering method, will serve as an example.

3.2.1 UPGMA

The "Unweighted Pair Group Method using Arithmetic means" groups those two taxa first whose evolutionary distance is minimal. Consider taxa A, B, C, and D with evolutionary distances as shown in Figure 9a. The taxa A and B with distance 6 are clustered first. Subsequently, A and B are treated as one compound taxon AB, and pairwise distances to the remaining taxa C and D are computed. $D_{(AB)C}$ is calculated as the arithmetic mean of the individual distances D_{AC} and D_{BC}, thus $D_{(AB)C} = (7 + 8)/2 = 7.5$. Likewise, we compute $D_{(AB)D} = (13 + 14)/2 = 13.5$. Now the cycle is repeated for the new 3×3 distance matrix. We obtain $((AB)C)$ and D as the two remaining

	A	B	C
B	6		
C	7	8	
D	13	14	11

a)



b)

Figure 9 Reconstruction of a phylogenetic tree with the UPGMA method. From the matrix of pairwise sequence distances (a) the phylogenetic tree shown in (b) is reconstructed. Numbers in (b) represent the branch lengths inferred under the assumption of a molecular clock. r identifies the root of the tree.

taxa with $D_{((AB)C)D} = (13 + 14 + 11)/3 = 12.7$. Finally, D and $((AB)C)$ are merged to conclude the procedure. The full tree $(((AB)C)D)$ with branch lengths is displayed in Figure 9b. Thus, UPGMA reconstructs a rooted tree, where branch lengths are computed such that the distances from root r to the leaves A, B, C, and D are identical (6.13 in our example). More generally, such trees fulfill the so-called ultrametric inequality, i.e. for each triple of taxa X, Y and Z:

$$D_{XY} \leq \max[D_{XZ}, D_{YZ}]. \tag{5}$$

Equation 5 is equivalent to the statement that two of the three distances are the same and at least as large as the third distance. More interestingly, the reverse is also true. If for a distance matrix the ultrametric inequality is fulfilled, then the distance matrix is representable by a rooted tree such that the distances D_{ij} are identical to the sum of the branch lengths connecting the two taxa X and Y in the tree. In biological parlance, if the distances computed from a set of aligned sequences obey the ultrametric inequality then the sequences evolve according to a molecular clock, i.e. they accumulate substitutions at the same rate (see Section 2). Therefore, UPGMA can give misleading trees if the distances reflect a substantial departure from the molecular clock. To arrive at a correct tree topology nevertheless, the distance matrix can be corrected for unequal rates of evolution among the lineages under study (*transformed distance method* [87]). The such modified distance matrix can then be used to infer the tree topology using UPGMA.

3.2.2 **Neighbors-relation Methods**

To overcome the restriction of the molecular clock, the characterization of unrooted trees with branch lengths is helpful. If it is required, alternative routes can be taken at a later step in the tree reconstruction to located the

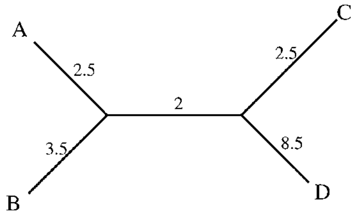


Figure 10 An unrooted quartet tree and its branch lengths reconstructed from the distance matrix in Figure 9(a).

position of the root (see Section 3.6). The celebrated *four-point condition* and its relaxations [5,20] state conditions when a distance matrix is representable as a tree. A distance matrix is representable as a tree if and only if for all quartets W, X, Y and Z in a taxon set the following holds:

$$D_{WX} + D_{YZ} \leq \max[D_{WY} + D_{XZ}, D_{WZ} + D_{XY}] \,. \tag{6}$$

The distance matrix in Figure 9(a) fulfills this criterion and the corresponding unrooted quartet tree is displayed in Figure 10. However, for real data the four-point condition is rarely met. Thus, one relaxes this condition by introducing the concept of *neighbors*. Two taxa are called neighbors in an unrooted tree if they are connected through a single internal node. For instance the taxa A and B in Figure 10 are neighbors, while the taxa A and C or B and D are not. This concept of neighborhood was generalized to distance matrices [5] and resulted in a series of tree reconstruction methods [5,51,131].

3.2.3 Neighbor-joining Method

A widely used method based on the neighbors-relation concept is the NJ method by Saitou and Nei [123]. NJ is a clustering algorithm. During each clustering step, two taxa or clusters of taxa are identified as neighbors in the tree, if their grouping results in a tree whose overall length is minimal, i.e. the sum of the lengths of all branches is minimal (minimum-evolution criterion [21]). To this end, one starts with a star-like tree. Subsequently, two taxa X and Y are identified that minimize:

$$S_{XY} = \frac{1}{2(N-2)} \sum_{k=3}^{N} (D_{Xk} + D_{Yk}) + \frac{1}{2} D_{XY} + \frac{1}{N-2} \sum_{3 \leq i \leq j \leq N} D_{ij} \,. \tag{7}$$

The cycle of calculating a new distance matrix and identifying the next neighbors is continued until the initially star-like tree is fully resolved (see also Section 12). For details of the NJ algorithm, see Ref. [147]. Since then, several weighted and improved versions of the NJ algorithm have been published [16,53].

3.2.4 **Least-squares Methods**

We have described the application of cluster methods to phylogeny reconstruction. However, another view of the reconstruction problem based on a distance matrix is the specification of an objective function we want to optimize. From a mathematical view, we want to find a tree together with its branch lengths such that the distance of two taxa X, Y in the tree, i.e. the sum of the branch lengths connecting the two taxa in the tree, is close to D_{XY}.

The least-squares method provides such a measure for the goodness of fit of the tree and its branch lengths to the data. The best tree (τ_{LS}) under this criterion minimizes the following equation:

$$R(\tau) = \sum_{XY} (T_{XY} - D_{XY})^2 \,, \tag{8}$$

where T_{XY} is the sum of the branch lengths along the unique path connecting sequences X and Y. Cavalli-Sforza and Edwards [21] and Fitch and Margoliash [49] were among the first to apply the least-squares theory to the tree-reconstruction problem. However, the big challenge is the determination of the tree topology.

3.3 The Criterion of Likelihood

The third method of tree reconstruction is based on the principle of ML [48] which was made popular in the field by Felsenstein in 1981 [42]. The general idea of ML is as simple as it is appealing: for a given model M and its parameters θ_M the probability or likelihood of observing data D can be calculated. Those parameters are chosen that maximize the likelihood of observing the data. For the particular problem of inferring a phylogenetic tree from biological sequence data the tree topology τ is introduced such that:

$$\tau_{\mathrm{ML}} = \underset{(\tau,\theta_M)}{\operatorname{argmax}} \, P(D|\tau, \theta_M). \tag{9}$$

Note the subtle, but far-reaching, difference to the principle of maximum parsimony. The general concept of sequence evolution inherent to maximum parsimony, i.e. that one sequence is transformed into another via the least number of changes, is replaced by an explicit model of sequence evolution to describe the substitution process. From this the most significant advantage of ML becomes apparent: it allows us to incorporate any model of biological sequence evolution into the tree reconstruction process. In this way, it opens access to the full use of statistical approaches to compare alternative phylogenetic hypotheses, as well as to test fit and robustness of individual models of sequence evolution. A further advantage compared to the previous two approaches is the possibility to make full use of the sequence information.

In a likelihood framework also constant alignment sites provide information about the tree topology and its branch lengths.

3.4 Calculating the Likelihood of a Tree

We have described in Section 2 how to calculate the probability of observing a difference at a given site in two sequences. We now extend this to compute the probability to find a certain nucleotide pattern (A_s) in column s of N aligned DNA sequences, e.g. the pattern `CCTC` at position 1 of the alignment shown in Figure 7. This probability depends on the model of DNA sequence evolution and on the tree relating the N nucleotides in the alignment column: $P(A_s|\tau, \theta_M)$. We assume that all positions in the alignment of length L evolve according to the same evolutionary model M and evolve independently from each other. Then, the probability of the alignment given a tree and a model is a function of the tree τ and θ_M. Thus:

$$P(A|\tau, \theta_M) = \prod_{s=1}^{L} P(A_s|\tau, \theta_M). \tag{10}$$

To avoid numerical problems caused by underflows and rounding errors during the calculation the likelihood of the data is usually calculated in log-scale, such that:

$$\log[P(A|\tau, \theta_M)] = \sum_{s=1}^{L} \log[P(A_s|\tau, \theta_M)]. \tag{11}$$

Equation (11) facilitates computation of the likelihood of an alignment, if θ_M, τ and its branch lengths are specified. In reality, however, we face the reverse situation. Starting from a given alignment, we aim to infer the underlying phylogenetic tree together with its branch lengths. In order to do so we regard these parameters as variables. Once we have decided on an evolutionary model and have specified its parameter values, we can adjust the tree topology and the branch lengths such that Eq. (11) is maximized. While straightforward and efficient ways exist to obtain ML branch lengths for a specific tree topology (e.g. Ref. [42]), it is a computationally demanding problem to obtain an optimal tree topology. Section 4 explains the details.

3.5 Bayesian Statistics in Phylogenetic Analysis

The likelihood approach outlined so far determines the quality of a tree by calculating the probability of observing the alignment A given the tree τ and the model of sequence evolution specified by θ_M (see Eq. 11). If we consider a particular combination of τ and θ_M as an evolutionary hypothesis, H, we have inferred $P(A|H)$, the probability of A given H. However, it might be interesting to address the reverse question: what is the probability that the evolutionary hypothesis H is correct given the data, i.e. $P(H|A)$ (see [37, p. 106])? Applying Bayes' theorem, we can calculate this posterior probability as:

$$P(H|A) = \frac{P(A|H)P(H)}{P(A)} . \tag{12}$$

Rewriting the equation for the problem of tree reconstruction we obtain:

$$P(\tau, \theta_M|A) = \frac{P(A|\tau, \theta_M)P(\tau, \theta_M)}{P(A)} , \tag{13}$$

where $P(\tau, \theta_M)$ is the prior probability to choose the tree τ and the model with its parameters. $P(A)$ is the total probability of the alignment A.

Equation (13) can be used in two ways for making phylogenetic inferences from a set of DNA sequences. If one is only interested in identifying the tree that is best supported by the data, one simply determines the tree and the θ_M that maximize Eq. 13. Because $P(A)$ is a constant it can be ignored during optimization. Alternatively, the posterior probability for every possible realizations of τ and θ_M (H_i) can be calculated. This identifies not only the H_i that is supported best by the data, but allows us also to assess how much better the support is compared to the alternative hypotheses [139]. However, it is easy to see that this is feasible for only a very limited number of sequences (see Section 4.1). Thus, Markov chain Monte Carlo (MCMC) simulations are used to estimate the posterior probabilities [105].

Imagine that the individual H_i comprise points in a landscape and $P(H_i|A)$ corresponds to their respective (unknown) altitude. A MCMC simulation is similar to a walk through this landscape that visits the individual points. This walk, however, is not totally random, but guided in a way that higher points are visited more often than lower ones. Thus, when the walk is finished an altitude profile of the landscape is generated from the number of times a particular point was visited. In practice, MCMC simulations work the following way. Starting from any H_i the transition to a new hypothesis H_j, e.g. a new tree topology, is proposed with a probability $q(H_i, H_j)$. This proposal is then accepted with probability:

$$\min\left(1, \frac{P(A|H_j)P(H_j)q(H_i, H_j)}{P(A|H_i)P(H_i)q(H_j, H_i)}\right) , \tag{14}$$

otherwise remaining at H_i. If H_j is supported better by the data than H_i, then H_j is always accepted. Otherwise, H_j is accepted with a probability that depends on how much worse the support of H_j is compared to H_i. The latter option ensures to escape from local optima (see Section 4). If the transition is accepted, H_j will be sampled and the chain moves on. Given that the chain could run for a sufficiently long time, the number of times H_i has been sampled reflects its posterior probability. However, in most cases it is not clear how long a chain should be run. To reduce potential biases of the Monte Carlo estimates, initial sample points are generally discarded [9]. This *burn-in* procedure has the effect that the chain samples only near-optimal hypotheses. Moreover, one samples only every 1000th hypothesis to generate more or less independent samples [46]

Today, more sophisticated MCMC simulations are performed that use several Markov chains whose "temperatures" differ [57, 77]. So-called "hot" chains have a high acceptance probability of inferior transition proposals. To stay in the above picture, they are used for a more global exploration of the landscape since their affinity to areas of high altitude is low. In contrast, "cold" chains with their low acceptance probability of inferior hypotheses are used for a thorough exploration of local areas in the landscape. Hypothesis sampling is done only from the cold chain. However, from time to time, the temperatures of the chains are swapped such that a hot chain is turned into a cold chain and *vice versa*. However, many more variants of MCMC sampling of phylogenies exist and the field is quickly evolving [33, 74–76, 95].

3.6 Rooting Trees/Molecular Clock

So far we have introduced various methods of inferring the relationships between sequences (or taxa). Unfortunately, most of the methods described above lack an inherent criterion for assigning directionality to the evolutionary process. As a consequence, they are unable to identify the root of a phylogenetic tree. To obtain a rooted tree, nevertheless, it is required (and possible) to add supplementary information into the tree reconstruction procedure.

3.6.1 **Outgroup Rooting**

Among the various methods for rooting a tree, it is most intuitive to divide the taxa into two subgroups: a monophyletic ingroup, i.e. taxa that share a common ancestor to the exclusion of all other taxa in the dataset, and an outgroup, whose more distant relationship to any member of the ingroup is either known or at least reasonable to assume (Figure 11a). It is then straightforward to conclude that the node that joins the outgroup to the ingroup represents the root of the ingroup subtree (r_{subtree} in Figure 11) [110, 140]. Though simple, this approach requires some considerations. Despite their clear position outside the ingroup, outgroup taxa should be as closely related to the ingroup taxa as possible. This will increase the probability of

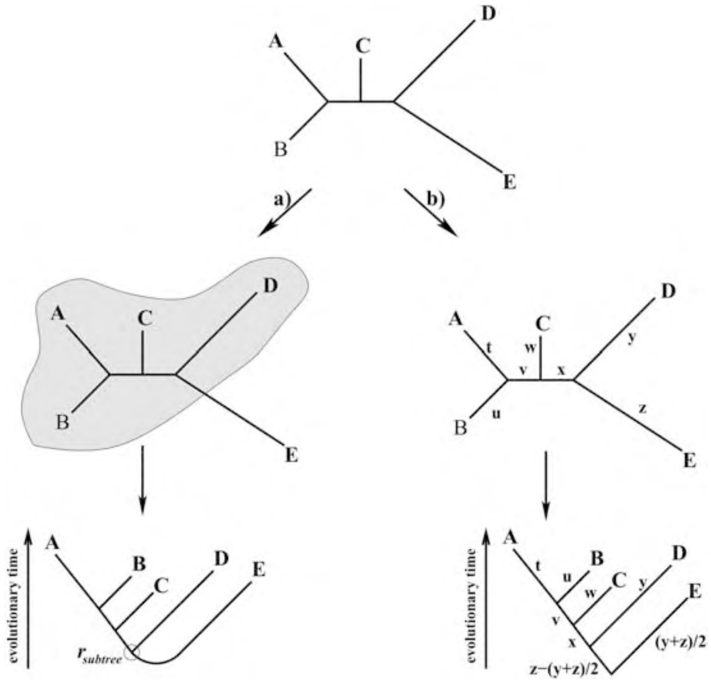


Figure 11 Two alternative principles for rooting phylogenetic trees. (a) Outgroup rooting. The set of taxa A–E is divided into an ingroup (shaded in grey) and the outgroup, taxon E. The node that joins the outgroup to the ingroup represents the root of the ingroup-subtree (r_{subtree}). The root of the entire tree, i.e. the common ancestor of all taxa, must be located somewhere on the outgroup branch. However, its exact position remains unknown. (b) Distance-based rooting. t to z denote the lengths of the individual branches in the tree. The root of the entire tree is identified as the midpoint of the path connecting the two taxa with the largest evolutionary distance.

reliably identifying homologous sequence positions using standard alignment procedures. Furthermore, it minimizes the risk of misplacing the outgroup due to its large evolutionary distance from the ingroup [99, 127, 154]. In addition to these more general requirements, some additional guidelines exist for rooting phylogenetic trees by an outgroup. First, more than one taxon should be included into the outgroup [100]. Furthermore, different outgroup taxa should be used to check whether the root placement depends on the choice of the outgroup [150].

3.6.2 **Midpoint Rooting and Molecular Clock**

As we have seen, the choice of a meaningful outgroup for rooting a phylogenetic tree can become a considerable problem. This is especially relevant when groups are analyzed whose phylogenetic relationships are unclear. In such cases additional assumptions about the evolutionary process are imposed that help rooting the tree.

Given that per unit of time any lineage accumulates the same amount of sequence changes (molecular clock) the point in the tree that is equally distant from all terminal taxa can be assigned as the root (see Section 3.2). In reality, however, the assumption of a molecular clock is frequently violated. If this is neglected rooting under the clock assumption tends to place the root in a part of the tree that is evolving at a high evolutionary rate. Midpoint rooting slightly relaxes the constraints imposed by the molecular clock assumption. It places the root on the midpoint of the path connecting the two most distantly related taxa in the phylogenetic tree (Figure 11b). Compared to the molecular clock scenario this retains only the postulation that the evolutionary rate has to be the same along the two most divergent lineages in the dataset. Midpoint rooting identifies the localization of the root correctly when this criterion is met [148].

4 Finding the Optimal Tree

So far we have outlined the principles to construct a phylogenetic tree from a set of aligned sequences. However, it still is unclear how to find the tree that reflects the relationships between the taxa best. We can differentiate between two general concepts of searching the tree space comprised by all possible tree topologies for the desired optimal tree: (i) *exhaustive* searches, which guarantee the identification of the optimal tree, and (ii) the computationally less-demanding *heuristic* searches that, however, do not necessarily obtain the globally optimal tree.

4.1 Exhaustive Search Methods

In the conceptually simplest approach, the exhaustive search, each and every possible bifurcating tree in the tree space is evaluated under the selected optimality criterion. The identification of the optimal tree(s) is then straightforward and the computational challenge is limited to exploring all of the tree space. To accomplish this, one starts with the (unique) unrooted tree that connects three randomly chosen taxa from the dataset. Subsequently, the remaining taxa are added in a step-wise fashion, such that the ith taxon is added separately to each of the $2i - 5$ branches of every possible tree for the $i - 1$ previous taxa. Obviously, the addition of every taxon increases the number of possible trees by the number of branches to which the new taxon can be connected [41]. Thus, the total number of unrooted trees for a set of n

taxa is:

$$B(n) = \prod_{k=3}^{n}(2k-5) = \frac{(2n-5)!}{2^{n-3}(n-3)!}. \tag{15}$$

The limitations of the exhaustive search are evident. Already a compilation of 20 taxa, a dataset that is nowadays easily exceeded, requires the evaluation of over 2×10^{20} different trees. This is, and presumably will remain, computationally infeasible.

Branch-and-bound methods [93] provide an alternative approach to finding a globally optimal solution without the need of evaluating all tree topologies. Instead, a guided search in the tree space is performed omitting those subspaces that cannot contain an optimal tree [67]. The rationale is simple and has the only prerequisite that the criterion of tree evaluation, i.e. the objective function F, is nondecreasing when new taxa are added to a particular subtree. If we want to minimize F, then we start with the computation of an upper bound F_{upper}, e.g., we evaluate any arbitrary n-taxon tree. Subsequently, using a three-taxon tree as a primer we recursively reconstruct the possible n-taxon trees. However, as we move along in our reconstruction procedure, i.e. with the addition of more and more taxa into the trees, we compare F of the resulting subtrees with F_{upper}. As soon as F_{subtree} exceeds F_{upper} we know that the search path leads to a subspace which contains trees where F is always larger that F_{upper}. Thus, no further reconstruction is required and another search path is evaluated. Alternatively, if we end up with a n-taxon tree, we store the new tree as candidate and update F_{upper} to the new value. The estimation of F_{upper} is crucial for the computational efficiency. Therefore, a number of improvements have been added to this basic scheme [67, 148]. These refinements are mainly designed to further reduce the exploration of tree space. They include methods for obtaining a near-optimal tree for an assessment of the initial upper bound, as well as schemes for generating a suitable order in which the taxa are added to the subtrees. For instance, by adding divergent taxa first, the length of the initial subtrees is increased, allowing for a quicker identification of subtrees that exceed the upper bound for the tree length.

Despite these improvements, exact searches eventually run into computational problems when data sets become large. For these cases, the considerably faster heuristic methods for tree reconstruction are required.

4.2 Heuristic Search Methods

Heuristic methods for tree reconstruction earn a substantial speed-up in computation time by jettisoning a guaranteed globally optimal solution to the tree search problem. With contemporary software it is possible to reconstruct trees from datasets of more than 1000 taxa (e.g. [63, 107, 141]). Thus, nowadays biological datasets hardly ever reach the computational limits of tree reconstruction software, provided that one is willing to abandon the guaranteed optimality.

4.2.1 Hill Climbing and the Problem of Local Optimization

The problem to find an optimal tree for a set of taxa can again be illustrated by the metaphor of exploring a landscape. A hiker aims to visit the point with the highest altitude in a hilly area. Due to the poor visibility, the highest peak cannot be identified *a priori*. Thus, the hiker remains with the only option to climb any slope he encounters first until he has reached its top. Up there he checks his altimeter and is either confident to have reached one of the highest points in this area and finishes his search or he invests more effort and climbs another hill. This kind of search strategy is called a local search.

Applying this approach to the tree search problem, we start off with any tree and modify it in a stepwise fashion, usually accepting only such modifications that result in an improved tree according to the chosen objective function. At a certain point no further improvement is possible and thus we have reached the top of the hill. At this point we have no means of deciding whether we have found the globally optimal tree or merely a local optimum and an optimization with a different initial tree would obtain better results. Consequently, tree searches based on a local search have to cope with three challenges: (i) the identification of a reasonable tree to start the search with, (ii) the implementation of a stepwise hill-climbing algorithm for the tree search and (iii) the avoidance of getting stuck in local optima that are highly suboptimal in terms of cost.

4.2.1.1 Identification of the Starting Tree

Reasonable starting trees are quickly obtained via so-called “greedy” strategies. The tree reconstruction is divided into several subproblems which are then sequentially solved by always choosing the solution that looks best given the current situation. In star decomposition methods (Figure 12), we begin with an assignment of all taxa to the terminal nodes of a star-like tree. Subsequently, all trees are evaluated that can be obtained by joining any two of the terminal taxa into a new group. The tree that scores best under the chosen optimality criterion forms the basis for the next step. The iteration of pairwise joining and tree evaluation continues until the tree is fully resolved.

Alternatively, we can directly construct a binary tree from scratch by inserting the taxa into a tree in a stepwise fashion (Figure 13) [38]. First, a set of three taxa is used to form a unique binary tree. Next, a fourth taxon is chosen for insertion into the initial tree. Since the taxon can be attached on any of the three branches of the initial tree, we have three possible topologies for the

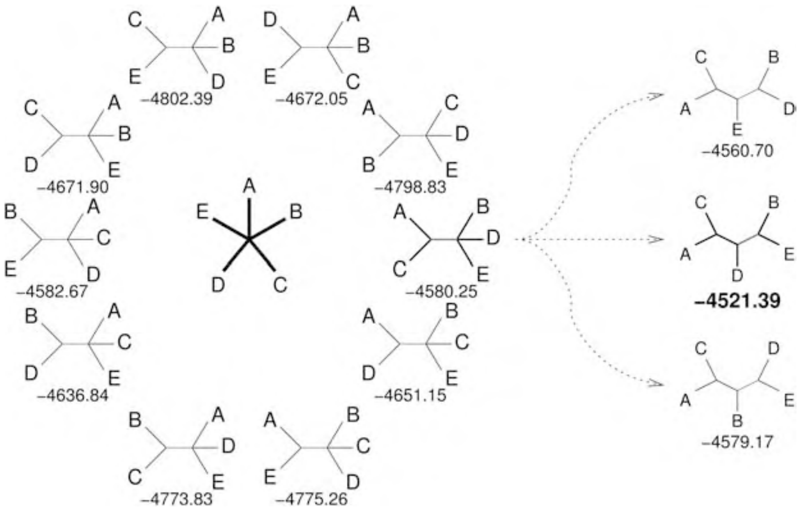


Figure 12 Star decomposition.

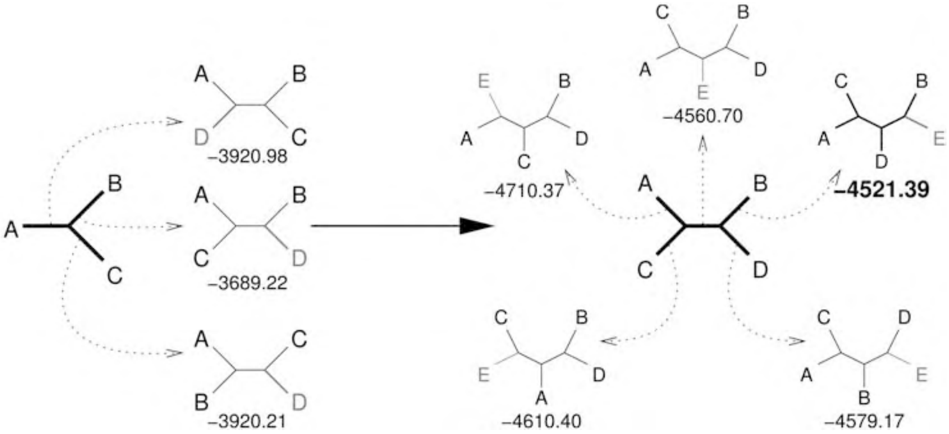


Figure 13 Stepwise insertion.

four-species tree. All of these will be evaluated and the best tree will be stored for insertion of the fifth taxon. The iteration continues until the tree includes all taxa in the dataset.

It is straightforward to see why both star decomposition methods as well as the stepwise insertion procedure are prone to obtaining only locally optimal trees. Any decision concerning the position of a taxon in the tree is fixed for the remaining part of the reconstruction procedure.

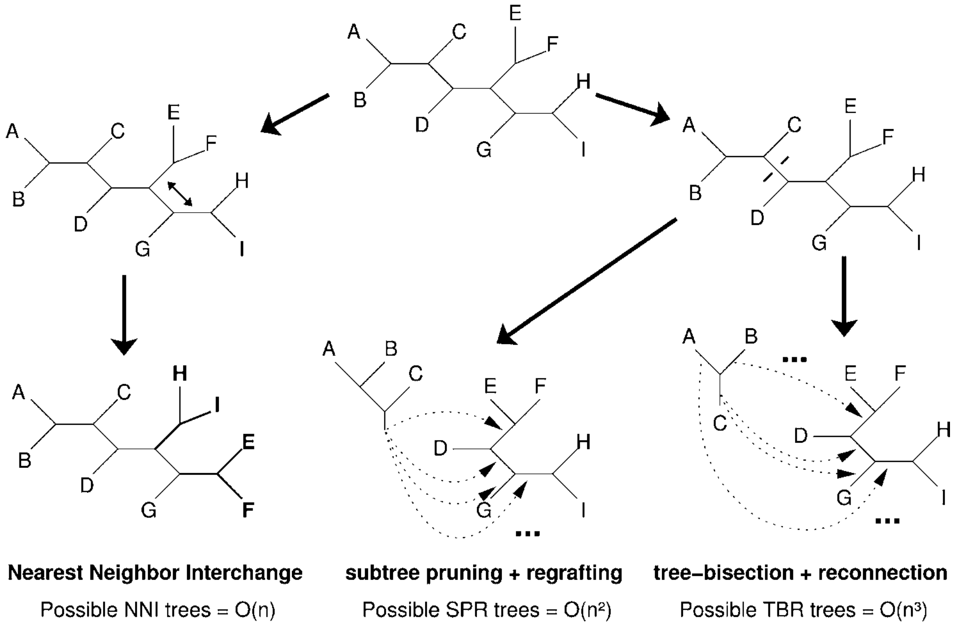

A
B
C
D
E
F
G
H
I
Nearest Neighbor Interchange
Possible NNI trees = O(n)
subtree pruning + regrafting
Possible SPR trees = O(n²)
tree-bisection + reconnection
Possible TBR trees = O(n³)

Figure 14 Three methods that accomplish branch swapping.

4.2.1.2 **Optimization Procedure and Avoidance of Local Optima**

In order to escape local optima, tree-rearrangement methods were suggested that override previous decisions concerning the placement of taxa in the tree. In brief, the initial "optimal" tree is modified such that a part of the tree is excised and re-inserted elsewhere. The trees resulting from such "branch swaps" are evaluated and subjected to one or more acceptance criteria. While a better tree is always accepted, trees inferior to the one already obtained can be accepted under certain conditions [77]. This deviation from the strict hill-climbing approach facilitates the transition to better trees that require more than one rearrangement of the current best tree. Note, the similarity to the MCMC approach (see Section 3.5). Currently, three branch-swapping methods are in use (Figure 14). Nearest neighbor interchange, the simplest approach, takes any internal branch of the tree and swaps two of the four connected subtrees. In this way, a total of $O(n)$ alternative trees are evaluated. (Note that only swapping two subtrees located on the opposite sides of the internal branch leads to the formation of a new tree!) Subtree pruning and regrafting ($O(n^2)$) excises a subtree and regrafts it with the cut surface at any branch on the tree. Tree bisection and reconnection is the most exhaustive way of swapping branches ($O(n^3)$). The tree is bisected along an internal branch and the resulting subtrees are rejoined at any pair of branches.

As noted, any of the branch-swapping methods is capable of guiding the tree-reconstruction procedure out of a local optimum. However, no guarantee is given that this does not simply lead into the next local optimum. Apparently, if the branch swapping is continued for a sufficient amount of time it becomes likely that sooner or later the global optimum will be found. However, how shall we recognize the globally optimal tree once we found it and for how long should we continue the tree search?

4.2.2 Modeling Tree Quality

It is inherent in the heuristic approach that, no matter how long we search, we can never be sure that we have found the globally optimal tree. Thus, we need a concept of tree quality, as we are continuing the search. In most cases it is essentially up to the user how long the search is continued. Either a predefined number of optimization steps or a lower limit by which new trees have to improve can be used as stopping criteria. However, both criteria are arbitrary and a well-founded basis for deciding when to end the search would be desirable. Recently, a method was suggested that is based on the rate of finding better trees during the search [152]. Let $F_1, F_2, \ldots, F_j$ denote the values of the objective function F for the trees found at iteration $1, 2, \ldots, j$. Then the sequence $r(k)$ of record times (i.e. number of iteration at which a better tree is found) is defined by:

$$r(1) = 1,\; r(k+1) = \min\{j | F_j > F_{r(k)}\}. \tag{16}$$

This sequence is used to estimate the point in time, r_{stop}, when to stop the search based on the probability of yet finding a better tree. Using the theory detailed in Refs. [23,120], one can estimate on the fly an upper 95% confidence limit $r_{95\%}$ of r_{stop}. Once $r_{95\%}$ iterations have been carried out and a better tree has not been detected the program will stop. It can then be concluded that with a probability of 95% no better tree will be found during this search. On the other hand, if a better tree is found before $r_{95\%}$ is hit, the $r_{95\%}$ is re-estimated on the basis of the new record time added to the sequence $r(k)$ and the search continues.

4.2.3 Heuristics for Large Datasets

The considerable ease with which DNA sequences are obtained nowadays results in ever-increasing datasets available for phylogeny reconstruction. As a consequence there is a demand for increasing the capacity of tree reconstruction software. One way to satisfy the needs is the development of parallelized versions of tree reconstruction programs, e.g. fastDNAml-based programs [111,141,142], TREE-PUZZLE [132], GAML [14] and MRBAYES [2].

The objectives for further improvements on the computational basis can be quickly summarized. (i) Finding in a shorter time a better starting tree for subsequent optimization. IQPNNI [152] accomplishes this by limiting the number of computation steps to place a new taxon during tree reconstruction. PhyNav [151], on the other hand, reduces the initial tree space by choosing for each group of closely related taxa one representative. From the resulting representative leaf set a scaffold is reconstructed, to which the initially deferred taxa are subsequently added such that an optimal tree is obtained. (ii) Improving the algorithms for tree optimization. For instance, PHYML [63] has implemented a fast algorithm for nearest-neighbor interchange, and RAxML [141] provides an improved version for subtree pruning and regrafting. (iii) The utilization of alternative approaches for tree reconstruction, such as a metapopulation genetic algorithm [96]. (iv) The dissection of the tree-reconstruction problem into a set of subproblems that can be solved on several CPUs in parallel. Some of these improvements are recent developments and it is not clear yet which combination will be optimal for tree reconstruction. In a sense an all-embracing optimal solution might be elusive since it is likely that different combinations will perform optimally on different data sets. Thus, it seems impossible to provide guidelines for when to use what program.

5 The Advent of Phylogenomics

A common problem for the accurate reconstruction of evolutionary relationships among taxa is the limited amount of phylogenetic signal in the data which, in addition, is frequently blanketed by noise. In view of the various genome sequencing efforts it seems trivial to enhance the signal-to-noise-ratio by the simple addition of more data [69]. However, even with the availability of whole-genome sequences, alignments remain limited to the level of individual genes in many cases. Both the rearrangement of genetic information in different taxa and the in part substantial sequence divergence of nonfunctional parts of the genome prevent the generation of meaningful longer sequence alignments. To extend the amount of information, nonetheless, disjoint datasets derived from multiple genomic loci can be combined for the analysis. This intersection of phylogenetics and genomics is referred to as phylogenomics.

5.1 Multilocus Datasets

Two approaches have been suggested for combining multilocus datasets from the same set of taxa for phylogenetics analysis. In supermatrix approaches [126] (also referred to as "total evidence" [89]) all individual sequence alignments are concatenated to form one large superalignment. The tree reconstruction is then based on this superalignment using standard methods. In

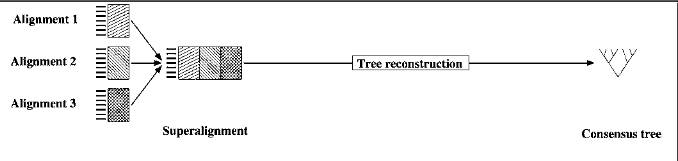
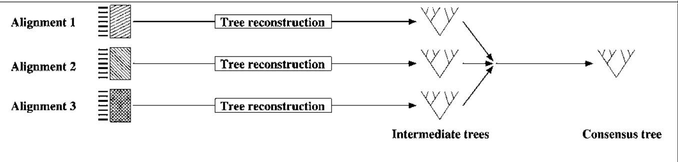
Combination level	Combination method	
early	*Supermatrix/ Total evidence*	
late	*Consensus tree*	

Figure 15 Two alternative methods to reconstruct a single phylogenetic tree from a set of disjoint alignments. In the early-level combination, the individual alignments represented by the patterned boxes are concatenated first to form a single superalignment. Standard phylogeny reconstruction programs can then be applied to reconstruct a tree from the superalignment. In the late-level combination a phylogenetic tree is reconstructed first for each alignment separately. The individual trees are later combined into a single consensus tree.

this approach, the phylogenetic information present in the individual alignments is combined early in the phylogenetic analysis. Hence, we refer to them as *early-level combination* methods (Figure 15, "early"). Alternatively, the information present in the individual alignments can be combined late in the phylogenetic analysis. Trees are reconstructed first for each alignment separately. These individual trees are then combined at a later step to form a so-called *consensus tree* (Figure 15 "late"). However, in contrast to the concatenation of individual alignments, which is simple text-editing, the combination of trees requires some further considerations.

A frequently used method for computing a consensus from a compilation of trees is based on the principle of identifying the set of compatible splits among these trees. To this end, splits comprise bipartitions of the taxon set that are induced by cutting a tree at any edge. More formally, splits are represented by the symbol "|" (Figure 16). Note that cutting at an external edge creates only trivial splits present in all trees. These are usually discarded from the analysis. Thus, we can induce for any tree with taxon set $\mathcal{N}$ a split $\mathcal{A}|\mathcal{B}$, such that $\mathcal{A} \cup \mathcal{B} = \mathcal{N}$ and $\mathcal{A} \cap \mathcal{B} = \emptyset$.

From the tree in Figure 16 we deduce the splits $\{A, B\}|\{C, D, E\}$ and $\{A, B, C\}|\{D, E\}$ (or shorter $AB|CDE$ and $ABC|DE$) We note that taxon C has changed sides. Thus, if we compute all four possible intersections between the splits only one will be empty. More formally, two splits $\mathcal{A}|\mathcal{B}$ and $\mathcal{C}|\mathcal{D}$ are said to be compatible if one of the four possible intersections $\mathcal{A} \cap \mathcal{C}$, $\mathcal{A} \cap \mathcal{D}$, $\mathcal{B} \cap \mathcal{C}$, $\mathcal{B} \cap \mathcal{D}$ is empty. If two intersections are empty the splits are identical. It is

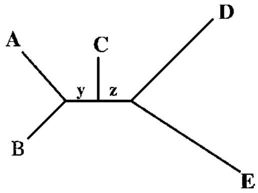


Figure 16 Two nontrivial splits can be derived from this tree. Cutting at the edge y induces the split $AB|CDE$. Cutting at edge z moves taxon C from the right-hand side of the split to the left-hand side and results in $ABC|DE$.

easy to see that splits derived from a tree are always pairwise compatible. On the other hand, a collection of splits that are pairwise compatible fit on a tree. Hence, collections of pairwise compatible splits are another way of encoding trees. For multilocus data the resulting trees are not necessarily the same and, thus, one needs approaches to summarize the results. The easiest form to summarize the result is simply counting the fraction ℓ at which a certain split occurs in a set of trees. If we collect only splits with $\ell > 50\%$, then the resulting system of splits is pairwise compatible and therefore representable as a tree [136] which we call M_{50} or majority rule consensus tree [102] (Figure 17). The cutoff value ℓ can of course be raised to construct more stringent majority consensus trees M_ℓ [102].

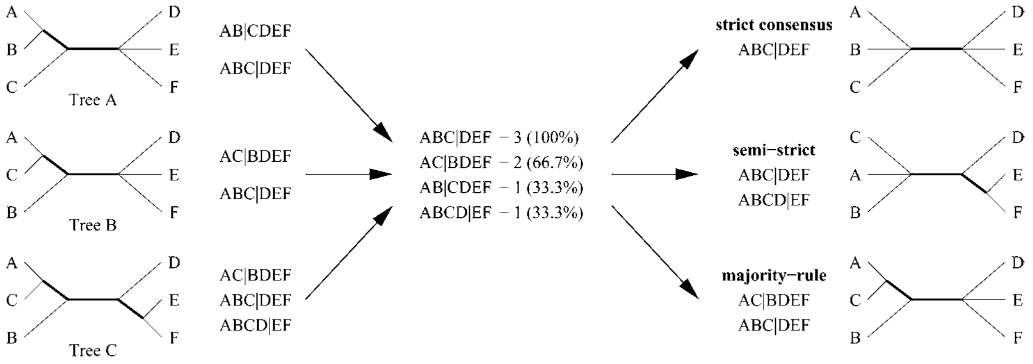


Figure 17 Examples for consensus methods to summarize a set of trees with identical taxon sets: strict (M_{strict}), semi-strict ($M_{\text{semi-strict}}$) and 50% majority rule consensus (M_{50}).

More restrictive cases of majority consensus are the strict consensus M_{strict} [122] that incorporates only splits present in all trees, and the semi-strict consensus $M_{\text{semi-strict}}$ [15] that contains all splits which are not contradicted by any split from the input trees (Figure 17). Many further methods exist for generating a consensus tree (e.g. Refs. [1,19,81,133]).

The application of consensus methods extends beyond the combination of trees from multilocus datasets. In principle, they can be used to summarize any set of trees, e.g., derived from Jackknife [106] or Bootstrap analysis [35, 44], sampled from MCMC simulations [77], or obtained by randomized input orders [144] to assess the reliability (or uncertainty) in the reconstructed trees.

5.2 Combining Incomplete Multilocus Datasets: Supertrees and their Methods

Consensus methods have one serious limitation – they are restricted to trees of equal size and taxon sets. The mutual coverage of currently available gene sequences and taxa is far from being satisfactory [32]. Thus, consensus methods are only applicable to very special and restricted multilocus data. This results in a trade-off between the number of taxa and the number of loci used in an analysis. Thus, one can study either many loci with only few taxa or vice versa. This situation will improve as more sequence data accumulate, especially in the wake of completely sequenced genomes. However, incomplete data will still remain simply because not all genes are present in the genomes of all taxa. Consequently, the question emerges of how to incorporate multiple incomplete datasets into phylogenetic analysis.

In principle, similar strategies are applicable as outlined in Section 5.1. Supermatrix methods use concatenated alignments. However, this requires that tree reconstruction methods must be able to handle the missing data. Simply discarding alignment positions with gaps would leave the user with only completely sampled loci or even no data at all.

When the data is combined at a late level in the analysis, several strategies are feasible. Methods have been proposed to combine separately reconstructed overlapping (typically rooted) trees of the different loci into one so-called supertree [11, 60] (Figure 18). Supertree approaches can be divided into two classes: agreement supertrees [12] and optimization supertrees [155].

5.2.1 Agreement Supertrees

Agreement supertree methods reconstruct a supertree based on those groupings that are shared or at least are uncontested among the set of rooted source trees [12]. This reflects the assumption that all source trees can in principle be obtained simply by pruning different sets of branches from one large tree, the parent tree, i.e. the source trees are compatible. It should, therefore, be straightforward to reconstruct the topology of the parent tree from the topologies of the source trees. Unfortunately, different parent trees may frequently lead to the same set of partial trees. In other words, agreement supertree reconstruction may result in different parent trees. The first supertree method available [60] was designed to find all possible parent trees

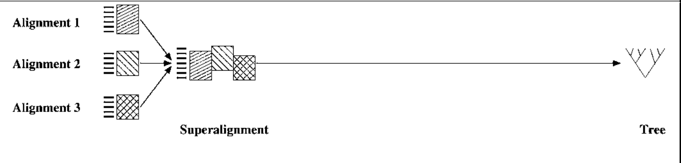
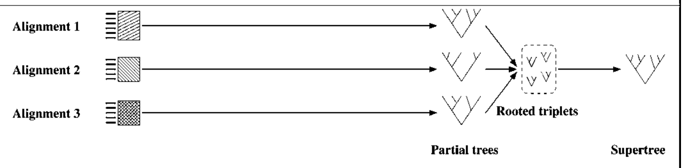
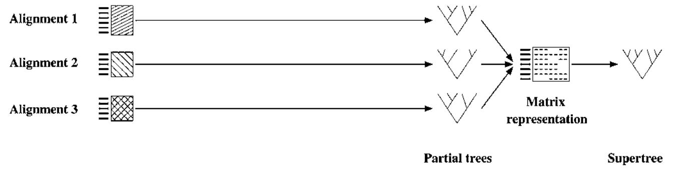
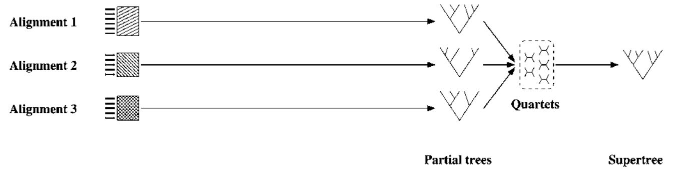
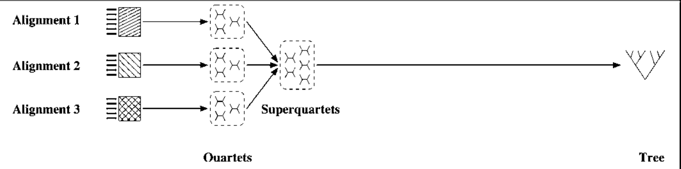
Combination level	Combination method	
early	*Supermatrix/ Total evidence*	 Alignment 1 Alignment 2 Alignment 3 Superalignment Tree
late	MinCut, ModMinCut	 Alignment 1 Alignment 2 Alignment 3 Rooted triplets Partial trees Supertree
late	*MRP, MRF*	 Alignment 1 Alignment 2 Alignment 3 Matrix representation Partial trees Supertree
late	*Quartet-based supertree*	 Alignment 1 Alignment 2 Alignment 3 Quartets Partial trees Supertree
medium	*Superquartet puzzling (SQP)*	 Alignment 1 Alignment 2 Alignment 3 Superquartets Ouartets Tree

Figure 18 A number of different methods to construct a single phylogenetic tree from a set of alignments with incomplete, but overlapping taxon sets. In early-level combination all alignments are concatenated into one large (super)alignment (or supermatrix, missing sequences are filled with gaps) from which the tree is reconstructed. In late-level combination the (typically rooted) trees are decomposed into sub-structures like rooted triplets (to obtain common nestings) or quartets, or are re-coded into a binary matrix representation (see Figure 19). These are then used to reconstruct a supertree. In medium-level combination with SQP the data is combined via quartets computed from each alignment. The resulting superquartets are then amalgamated into an overall tree.

for a set of partial trees and compute the strict consensus from the different parent (see Section 5.1). The resulting supertree, however, displays only those bipartitions that are supported in all parent trees, but some information about the structure present only in a fraction of the parent trees might be concealed.

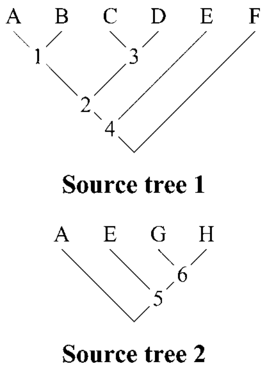
Thus, subsequent approaches like OneTree [17] returned only a single possible parent tree as the supertree. Obviously, the strict requirement of the source trees being compatible severely limits the applicability of these supertree methods. Any real application leads to source trees that are incompatible for a variety of reasons, one of which is just by chance. Thus, subsequent agreement supertree methods, such as the MinCut Supertree algorithm [135] or the ModMinCut Supertree algorithm [112] aimed to overcome the requirement of compatibility. In brief, they introduced a weighting of the links between taxa (i.e. common occurrence in the same subtrees) during the reconstruction of supertrees, such that the weight of a link increases the more source trees display this link. Subsequently, if a subtree cannot be resolved further due to incompatible source trees, the subtree is resolved by pruning those links with the lowest weight (MinCut) greedily. Furthermore, the ModMinCut Supertree algorithm [112] aims to keep links that are uncontradicted, even if they are established only by a single input tree, which would cause the MinCut Supertree algorithm to discard it. Further agreement supertree methods have been suggested recently and a comprehensive overview is given in Refs. [10,12].

5.2.2 **Optimization Supertrees**

Here, the set of input trees is decomposed into smaller entities. These entities serve as input to reconstruct an overall tree based on an objective function.

Matrix representation methods are one example. Prior to constructing the supertree, the rooted input trees are encoded into a binary matrix. Typically each internal node in the (rooted) input trees is encoded either by its adjacent subtree (Ragan/Baum scheme [7, 119], Figure 19a) by assigning "1" to taxa within the subtree and "0" otherwise, or its adjacent sister groups (Purvis' scheme [118], Figure 19b) assigning "0" to the taxa in one sister group and "1" to the other. All other and missing taxa of the tree are assigned "?". The obtained matrix representation of the input trees is then used as input alignment to reconstruct a supertree (Figure 18). For this purpose, various optimizing algorithms can be applied, such as (i) parsimony (MRP [7, 119]), which is to date the by far most common method, (ii) distance-based methods (MRD or average consensus method [94]), and (iii) finding the optimal tree which requires the least changes between ones and zeros (flips) to be congruent with the matrix (MRF [22]).

As an alternative to the matrix representation method, quartet-based supertree methods make direct use of the topological information in the input trees. To this end, the source trees are decomposed into quartet trees, which then serve as building blocks to reconstruct of the supertree [115, 121] (Figure 18).



a) Ragan/Baum

	inner nodes					
taxa	1	2	3	4	5	6
A	1	1	0	1	0	0
B	1	1	0	1	?	?
C	0	1	1	1	?	?
D	0	1	1	1	?	?
E	0	0	0	1	1	0
F	0	0	0	0	?	?
G	?	?	?	?	1	1
H	?	?	?	?	1	1

b) Purvis

	inner nodes					
taxa	1	2	3	4	5	6
A	1	1	0	1	0	**?**
B	1	1	0	1	?	?
C	0	1	1	1	?	?
D	0	1	1	1	?	?
E	**?**	0	**?**	1	1	0
F	**?**	**?**	**?**	0	?	?
G	?	?	?	?	1	1
H	?	?	?	?	1	1

Figure 19 For matrix representation methods like MRP a set of rooted input trees are encoded into a binary matrix. Each internal node (here 1–6) forms a column in a "binary" matrix. In the coding scheme of (a) Baum and Ragan a taxon that is located in the subtree associated to the current internal node is assigned "1", and a "0" otherwise. Missing taxa get the character "?". (b) In Purvis' scheme only adjacent sister groups are encoded to compensate for different tree sizes. Taxa connected to one subtree are assigned "1", those of the sister subtree "0". Missing taxa and those located root-wards are assigned the character "?".

5.2.3 **The Supertrees/Consensus versus Total Evidence Debate**

Alternative approaches to reconstruct trees from incomplete multilocus data ultimately invoke the debate whether supertree/consensus methods are superior to a supermatrix approach [10,13,24,108], or *vice versa* [34,55,89,90]. Meanwhile a number of points of critique have been raised against either approach, as have advantages. The predominant critique on the supermatrix (total evidence) approach addresses the issue of choosing an evolutionary model and its parameters such that the various evolutionary constraints imposed on the different concatenated datasets are reflected. On the other hand, supertree methods are criticized for their careless treatment of information provided by the data. Usually, the underlying sequence data is discarded prior to combining the input trees. Thus any information not represented by the tree topology is inevitably lost. Furthermore, supertrees carry the risk of possible unwanted data duplication and weighting [54,56], especially if tree topologies have been collected from the literature.

5.2.4 **Medium-level Combination**

Based on the above criticisms, a third level of dataset combination has been proposed recently [133], which takes an intermediate position between the (late) supertree and the (early) supermatrix approaches. Thus, we call it medium-level combination. The so-called superquartet puzzling algorithm (SQP [133]) combines the data on the level of four-taxon (quartet) trees. These so-called superquartets are then used as building blocks in the reconstruction using a voting scheme and a construction algorithm that is aware of missing data. SQP allows the application of evolutionary models fitted to each locus separately. Furthermore, it uses the phylogenetic information in the sequences both for combining the data and in the reconstruction of the final tree. Thus, SQP tries to combine known advantages of supermatrix and supertree methods. In addition, SQP can use datasets without defined outgroups or other information where the root has to be placed, which is generally required by most other supertree methods.

6 Phylogenetic Network Methods

By the nature of the biological data, the tree-reconstruction methods presented so far are only approximate methods. For any sufficiently large dataset, the four-point condition (Eq. 6) introduced in Section 3.2 is frequently not fulfilled. In such cases either the assumption of the underlying data evolving according to a (single) tree is not valid or methodological shortcomings disguise the tree-like evolution of the data. A number of reasons why the evolutionary relationships of DNA sequences might not resemble a tree (both methodological and biological) have been outlined earlier in this chapter (see Sections 1.2.1 and 5.1). Irrespective of the cause for the nontree-likeness of the phylogenetic signal in the data, it is obvious that coming up with a single tree is a feature of tree-reconstruction methods that might not be always desirable.

In the case of conflicting evolutionary signals in the data, a tree might not be the appropriate form to representing the phylogenetic relationships for a set of sequences. Thus, a series of algorithms have been proposed that are useful additions a to tree-based analysis. These methods can visualize to some extent conflicting alternative taxon groupings that cannot be represented by a single tree [6]. Nowadays, such algorithms are subsumed under the notion of (phylogenetic) network methods [116].

6.1 From Trees to Split Networks

6.1.1 Split Systems and their Visualization

Recall the idea of representing a tree by a set of *splits* introduced in Section 5.1. By definition, splits derived from a single tree are always compatible and, in turn, a tree can be reconstructed from a set of compatible splits. In order to combine phylogenetic information present in a set of trees based on sequences from various genetic loci the collection of splits observed in the individual trees can be collected and analyzed. Usually, not all splits are compatible and thus *strict* or *majority-rule consensus* trees are applied to filter the set of splits prior to tree reconstruction.

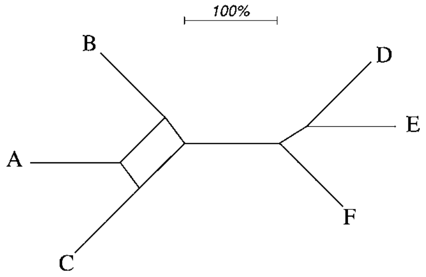


Figure 20 Consensus network representing all four splits collected from the three input trees in Figure 17. Branch lengths are drawn relative to the percent occurrence in the input trees. The compatible splits $ABC|DEF$ and $ABCF|DE$ form the tree-like branches in the right part of the graph, while the pairwise incompatible splits $AB|CDEF$ and $AC|BDEF$ form the net structure on the left. Note that the contraction of one set of parallel branches each obtains the corresponding tree responsible for the incompatible split.

One way to visualize incompatible splits present in the data goes along the lines with these consensus methods. Instead of stopping at the 50% cutoff which guarantees that the outcome is, in fact, a tree, one keeps adding less frequent splits obtained from a set of input trees to the split system, i.e. the set of splits. More generally, the application of a cutoff value r (analogous to ℓ in consensus trees) allows the selection of any split system S_r that is present in at least a portion r of all input trees. Pushing r below 50% ($r \leq 0.5$) may lead to a splits system that no longer conforms to a tree. Visualizing such incompatible splits systems provides insights into the extent and pattern of heterogeneity of the phylogenetic signal in the data. In such a network, one split is represented either by a single branch or by parallel branches, indicating incompatible splits as in Figure 20, where $AC|BDEF$ and $AB|CDEF$ are incompatible.

It has been shown (see Ref. [73]), that a split system S_r with cutoff fraction r does contain any subset larger than $\lfloor 1/r \rfloor$ splits which are all pairwise incompatible. S_r is said to be $(\lfloor 1/r \rfloor)$-compatible. The split system in Figure 20, for example, is 2-compatible, containing the subset of two pairwise incompatible splits $AC|BDEF$ and $AB|CDEF$. All split systems S_r with $r > 0.5$ are 1-compatible, which means that there are no incompatible splits and, hence, the resulting topology would again be a tree.

The amount of pairwise incompatible splits obviously determines the complexity of the network containing them. Median networks [4, 72] can contain cubes of dimension up to $\lfloor 1/r \rfloor$, and might thus be utterly complex. For example, a split system $S_{0.25}$ can be 4-compatible and, hence, needs four dimensions to be visualized.

Median networks are a very general type of network which can be reconstructed from the binary encoding of a split system. To this end, for each split taxa on one side of the split are assigned ones, those on the other side zeros. Then, intermediate states (representing the inner nodes in the median network) are computed from the binary sequences in a parsimonious fashion (see Section 3.1). It has been shown [4,72] that by pruning branches from a median network one can extract all the most parsimonious trees for the split system. Due to the fact that median networks can grow arbitrarily incomprehensible, less-complex approximations are often applied. Split graphs [6, 30, 31], for example, attempt to filter the splits and branches drawn, to derive a planar graph, i.e. a graph without intersecting edges. Refer to Ref. [72] for a more detailed overview.

6.1.2 Constructing Split Systems from Trees

Commonly, split systems are collected from a set of input trees with equal taxon sets as they are obtained, for example, from Bootstrap analysis or MCMC sampling (see Section 3.5). Similar to consensus trees (see Section 5.1), such split systems are visualized as so-called *consensus networks* [70]. Such consensus networks (see Figure 20) visualize the area and the extent of contradiction of the phylogenetic signal found in the input trees. However, like supertrees (Section 5.2), network reconstruction is not restricted to trees with equal taxon set, but can also be done from overlapping trees using the Z-closure method [78]. In accordance to the amalgamation of trees to supertrees, such networks are then called “super-networks” (Note, that the “super” prefix in super-networks does not follow the same notation as in supertrees, super-alignments, or superquartets, since is not network constructed from networks, but from trees.)

6.1.3 Constructing Split Systems from Sequence Data

Although applications such as consensus networks and super-networks were suggested quite recently, one should note that the basic idea of representing evolutionary processes by networks rather than trees is not new.

In contrast to approaches for network reconstruction based on collections of splits derived from trees, distance-based methods including split decomposition [6] or Neighbor-Net [18] have been suggested for constructing split systems directly from the data.

Applied to viral sequences from Ref. [124], methods like split decomposition and Neighbor-Net can easily identify the presence of the three recombinant HIV strains SE7812_2, UG266 and VI1310-1.7 (Figure 21b and c). These would have been wrongly classified based on the result of a tree reconstruction method like BioNJ (Figure 21a). To avoid such missinterpretation of the data, phylogenetic networks should be taken into account at early stages of the analysis.

Programs like SplitsTree [80], T-REX [101] NETWORK [3] and Spectronet [72] provide easy to use means that can be readily applied to a phylogenetic

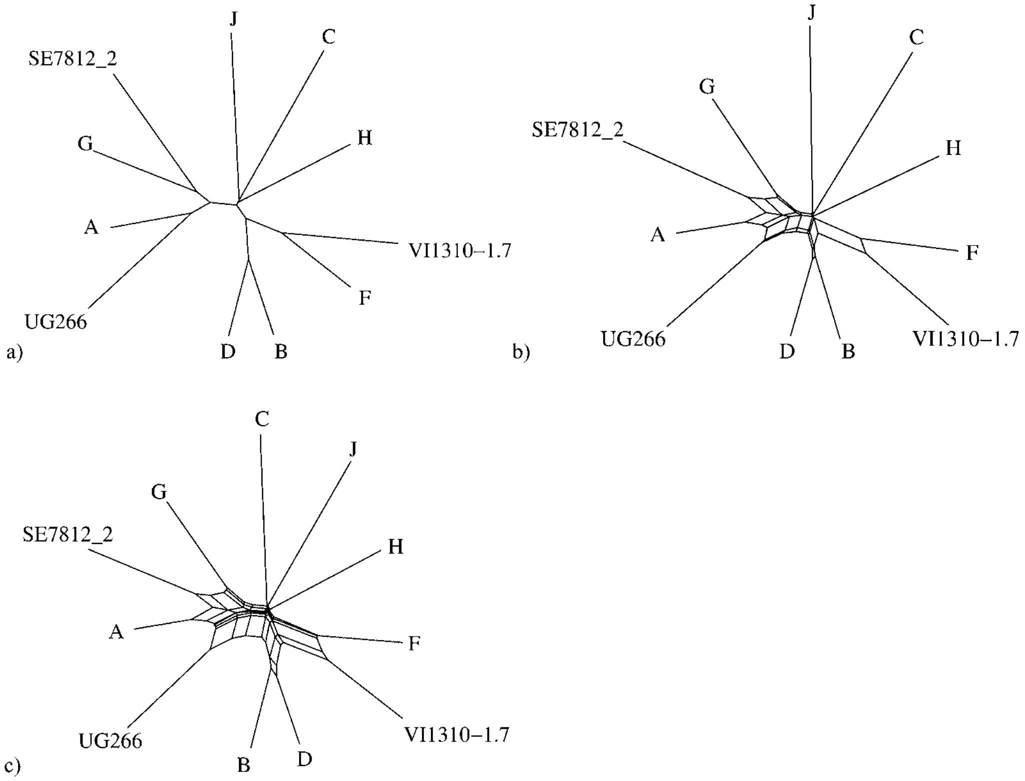


Figure 21 Results for a set of HIV dataset from Ref. [124] containing reference strains (A, B, C, D, F, G, H and J) together with three recombinants (SE7812_2, UG266 and VI1310-1.7). (a) BioNJ tree (eight splits), (b) split decomposition (14 splits) and (c) Neighbor-Net (19 splits).

analysis. These packages were used to analyze viral data [29, 124], hybridization events [98] and intra-specific data [4], respectively.

It should be noted that such split-based networks provide only a visualization of ambiguities in the data and do not qualify as methods to infer the reasons for the net-like structure [18].

6.2 Reconstructing Reticulate Evolution and Further Analyses

As mentioned above, in the case that different methods, different loci or even just different parts of the very same gene show conflicting phylogenetic signals, various causes might account for the observed conflicts.

One would certainly first check whether the conflicting evolutionary hypotheses are really significantly different [58, 66, 138]. If so, we can envi-

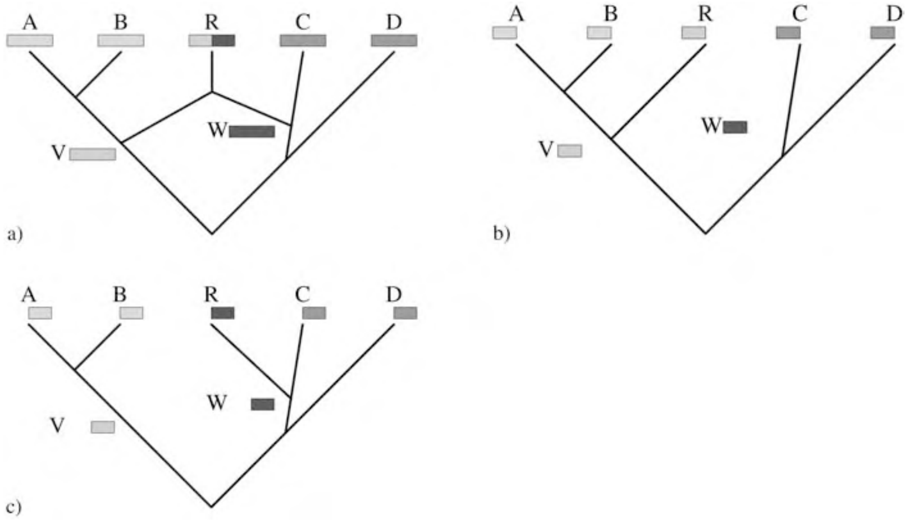


Figure 22 A case of reticulate evolution. (a) The recombination between strain V and W forms the recombinant strain R. The two parts of the sequence reflect different evolutionary histories (b) and (c) of the reticulation.

sion several biological mechanisms to produce reticulations, i.e. network-like evolution. In the particular example of virus evolution, reassortment, i.e. the mixture of viral chromosomes within a cell co-infected by different viral lineages [86], and recombination, i.e. the reciprocal exchange of genomic regions among chromosomes [113, 117] both of which are highly abundant in viruses [52, 125], are two examples. Horizontal gene transfer [8, 25, 62] and genome hybridization or fusion [28, 103, 158] constitute other possibilities. Sometimes reticulations may be simply due to parallel substitutions in different organisms. In such a situation the loops in the network indicate the occurrence of reverse or parallel mutations.

In general, purely split-based networks are not sufficient to illustrate an event of reticulate evolution like the recombination shown in Figure 22, because one cannot represent this evolutionary history as a set of splits, although one could map the separate gene trees. Hence, recently network methods like hybridization networks, recombination networks or galled trees [64, 79] have been suggested to reconstruct reticulate evolution.

The pros and cons of the different algorithms still need to be evaluated. However, the comparison of different phylogenetic networking strategies is not easy. From Figure 21 it is apparent that Neighbor-Net is more liberal in introducing noncompatible splits (19 splits in the Neighbor-Net compared to 14 splits in the split decomposition network and eight splits in a fully resolved tree.) Simulation studies, successfully applied in phylogenetic inference, might be one way to evaluate accuracy and robustness of phylogenetic network methods. To this end a carefully designed experimental setup is required. An alternative approach is the introduction of optimality criteria (least-squares, likelihood) in the network construction process. Unfortunately, the development of phylogenetic networks in a likelihood framework is still in its infancy [143, 146, 153]. Only with such methods is one able to decide whether the nontree-like signals are biologically plausible or are merely an artifact of the reconstruction procedure.

In this chapter, we have only touched upon some phylogenetic network methods. Space limitations do not allow a full account of all existing methods. The pyramidal clustering technique [26] is, like Neighbor-Net, an agglomerative approach. Statistical geometry in sequence space [36] and its descendants [109, 145] are alternative approaches to summarize the extent of tree-likeness in the data without reconstructing phylogenetic networks.

The field of phylogenetic networks will certainly profit from the data produced in whole-genome projects. As independently evolving DNA segments of eukaryotic genomes may display different evolutionary histories, the correct history of the taxa, as carriers of these segments, is probably more adequate. However, before we can reliably do this, we need to distinguish true loops in a network from artificial loops generated by too simple assumptions about the evolutionary process.

References

1 ADAMS III, E. N. 1986. $N-$trees as nestings: complexity, similarity, and consensus. J. Classif. **3**: 299–317.

2 ALTEKAR, G., S. DWARKADAS, J. P. HUELSENBECK AND F. RONQUIST. 2004. Parallel Metropolis coupled Markov chain Monte Carlo for Bayesian phylogenetic inference. Bioinformatics **20**: 407–15.

3 BANDELT, H.-J., P. FORSTER AND A. RÖHL. 1999. Median-joining networks for inferring intraspecific phylogenies. Mol. Biol. Evol. **16**: 37–48.

4 BANDELT, H.-J., P. FORSTER, B. C. SYKES AND M. B. RICHARDS. 1995. Mitochondrial portraits of human populations using median networks. Genetics **141**: 743–53.

5 BANDELT, H.-J. AND A. W. M. DRESS. 1986. Reconstructing the shape of a tree from observed dissimilarity data. Adv. Appl. Math. **7**: 309–43.

6 BANDELT, H.-J. AND A. W. M. DRESS. 1992. Split decomposition: a new and useful approach to phylogenetic analysis of distance data. Mol. Phylogenet. Evol. **1**: 242–52.

7 BAUM, B. R. 1992. Combining trees as a way of combining data sets for phylogenetic inference, and the desirability of combining gene trees. Taxon **41**: 3–10.

8 BERGTHORSSON, U., K. L. ADAMS, B. THOMASON AND J. D. PALMER. 2003. Widespread horizontal transfer of mitochondrial genes in flowering plants. Nature **424**: 197–201.

9 BESAG, J. AND P. J. GREEN. 1993. Spatial statistics and bayesian computation (with discussion). J. R. Statist. Soc. B **55**: 25–37.

10 BININDA-EMONDS, O. R. P., K. E. JONES, S. A. PRICE, M. CARDILLO, R. GRENYER AND A. PURVIS. 2004. Garbage in,

garbage out: data issues in supertree construction. In BININDA-EMONDS O. R. P. (ed.), *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life.* Kluwer, Dordrecht: 267–80.

11 BININDA-EMONDS, O. R. P. 2004. *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life.* Kluwer, Dordrecht.

12 BININDA-EMONDS, O. R. P. 2004. The evolution of supertrees. TREE **19**: 315–22.

13 BININDA-EMONDS, O. R. P. 2004. Trees versus characters and the supertree/supermatrix "paradox". Syst. Biol. **53**: 360–1.

14 BRAUER, M. J., M. T. HOLDER, L. A. DRIES, D. J. ZWICKL, P. O. LEWIS AND D. M. HILLIS. 2002. Genetic algorithms and parallel processing in maximum-likelihood phylogeny inference. Mol. Biol. Evol. **19**: 1717–26.

15 BREMER, K. 1990. Combinable component consensus. Cladistics **6**: 369–72.

16 BRUNO, W. J., N. D. SOCCI AND A. L. HALPERN. 2000. Weighted neighbor joining: a likelihood-based approach to distance-based phylogeny reconstruction. J. Mol. Evol. **17**: 189–97.

17 BRYANT, D. AND M. STEEL. 1995. Extension operations on sets of leaf-labeled trees. Adv. Appl. Math. **16**: 425–53.

18 BRYANT, D. AND V. MOULTON. 2004. Neighbor-Net: an agglomerative method for the construction of phylogenetic networks. Mol. Biol. Evol. **21**: 255–65.

19 BRYANT, D. 2003. A classification of consensus methods for phylogenetics. Bioconsensus: Proc. of Tutorial and Workshop on Bioconsensus II DIMACS-AMS, Piscataway, NJ: 55–66.

20 BUNEMAN, P. 1971. The recovery of trees from measurements of dissimilarity. In HODSON F. R., D. G. KENDALL AND P. TAUTU (eds.), *Mathematics in the Archeological and Historical Sciences*. Edinburgh University Press, Edinburg: 387–95.

21 CAVALLI-SFORZA, L. L. AND A. W. F. EDWARDS. 1967. Phylogenetic analysis: models and estimation procedures. Evolution **21**: 550–70

22 CHEN, D., L. DIAO, O. EULENSTEIN, D. FERNÁNDEZ-BACA AND M. J. SANDERSON. 2003. Flipping: a supertree construction method. In JANOWITZ M. F., F.-J. LAPOINTE, F. R. MCMORRIS, B. MIRKIN AND F. S. ROBERTS (eds.), *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* American Mathematical Society, Providence, RI: 135–60.

23 COOKE, P. 1980. Optimal linear estimation of bounds of random variables. Biometrika **67**: 257–8.

24 DE QUEIROZ, A., M. J. DONOGHUE AND J. KIM. 1995. Separate versus combined analysis of phylogenetic evidence. Annu. Rev. Ecol. Syst. **26**: 657–81.

25 DELWICHE, C. F. AND J. D. PALMER. 1996. Rampant horizontal transfer and duplication of rubisco genes in eubacteria and plastids. Mol. Biol. Evol. **13**: 873–82.

26 DIDAY, E. AND P. BERTRAND. 1986. An extension of hierarchical clustering: the pyramidal representation. In GELSEMA, E. S. AND KANAL, L. N. (eds.), *Pattern recognition in Practice*. North-Holland, Amsterdam: 411–24.

27 DOBZHANSKY, T. 1973. Nothing in biology makes sense except in the light of evolution. Am. Biol. Teach. **35**: 125–29.

28 DOOLITTLE, W. F. 1999. Phylogenetic classification and the universal tree. Science **284**: 2124–8.

29 DOPAZO, J., A. DRESS AND A. VON HAESELER. 1993. Split decomposition: a technique to analyze viral evolution. Proc. Natl. Acad. Sci. USA **90**: 10320–4.

30 DRESS, A., D. HUSON AND V. MOULTON. 1996. Analyzing and visualizing sequence and distance data using SplitsTree. Discr. Appl. Math. **71**: 95–109.

31 DRESS, A. W. M. AND D. H. HUSON. 2004. Constructing splits graphs. IEEE/ACM Trans. Comput. Biol. Bioinform. **1**: 109–15.

32 DRISKELL, A. C., C. ANÉ, J. G. BURLEIGH, M. M. MCMAHON, B. C. O'MEARA AND M. J. SANDERSON. 2004. Prospects for building the tree of life from large sequence databases. Science **306**: 1172–4.

33 DRUMMOND, A., A. RAMBAUT, B. SHAPIRO AND O. G. PYBUS. 2005.

Bayesian coalescent inference of past population dynamics from molecular sequences. Mol. Biol. Evol. **22**: 1185–92.

34 Eernisse, D. and A. G. Kluge. 1993. Taxonomic congruence versus total evidence, and amniote phylogeny inferred from fossils, molecules, and morphology. Mol. Biol. Evol. **10**: 1170–95.

35 Efron, B. 1979. Bootstrap methods: another look at the Jackknife. Ann. Stat. **7**: 1–26.

36 Eigen, M., R. Winkler-Oswatitsch and A. Dress. 1988. Statistical geometry in sequence space: a method of quantitative comparative sequence analysis. Proc. Natl. Acad. Sci. USA **85**: 5913–7.

37 Ewens, W. J. and G. R. Grant. 2001. *Statistical Methods in Bioinformatics: An Introduction.* Springer, New York, N.Y.

38 Farris, J. S., A. G. Kluge and M. J. Eckhardt. 1970. A numerical approach to phylogenetic systematics. Syst. Zool. **19**: 172–89.

39 Farris, J. S. 1977. Phylogenetic analysis under Dollo's Law. Syst. Zool. **26**: 77–88.

40 Farris, J. S. 1996. Parsimony jackknifing outperforms neighbor-joining. Cladistics **12**: 99–124.

41 Felsenstein, J. 1978. The number of evolutionary trees. Syst. Zool. **27**: 27–33.

42 Felsenstein, J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. **17**: 368–76.

43 Felsenstein, J. 1984. Distance methods for inferring phylogenies: a justification. Evolution **38**: 16–24.

44 Felsenstein, J. 1985. Confidence limits on phylogenies: an approach using the bootstrap. Evolution **39**: 783–91.

45 Felsenstein, J. 1987. Estimation of hominoid phylogeny from a DNA hybridization data set. J. Mol. Evol. **26**: 123–31.

46 Felsenstein, J. 2004. *Infering Phylogenies.* Sinauer, Sunderland, MA.

47 Ferris, S. D., A. C. Wilson and W. M. Brown. 1981. Evolutionary tree for apes and humans based on cleavage maps of mitochondrial DNA. Proc. Natl. Acad. Sci. USA **78**: 2432–6.

48 Fisher, R. A. 1912. On an absolute criterion for fitting frequency curves. Mess. Math. **41**: 155–60.

49 Fitch, W. M. and E. Margoliash. 1967. Construction of phylogenetic trees. Science **155**: 279–84.

50 Fitch, W. M. 1971. Toward defining the course of evolution: Minimum change for a specific tree topology. Syst. Zool. **20**: 406–16.

51 Fitch, W. M. 1981. A non-sequential method for constructing trees and hierarchical classifications. J. Mol. Evol. **18**: 30–7.

52 Fitch, W. M. 1996. The variety of human virus evolution. Mol. Phylogenet. Evol. **5**: 247–58.

53 Gascuel, O. 1997. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. Mol. Biol. Evol. **14**: 685–95.

54 Gatesy, J., C. Matthee, R. DeSalle and C. Hayashi. 2002. Resolution of a supertree/supermatrix paradox. Syst. Biol. **51**: 652–64.

55 Gatesy, J., R. H. Baker and C. Hayashi. 2004. Inconsistencies in arguments for the supertree approach: supermatrices versus supertrees of *Crocodylia*. Syst. Biol. **53**: 342–55.

56 Gatesy, J. and M. S. Springer. 2004. A critique of matrix representation with parsimony supertrees. In Bininda-Emonds, O. R. P. (ed.), *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*. Kluwer, Dordrecht: 369–88.

57 Geyer, C. J. 1991. Markov chain Monte Carlo maximum likelihood. Proc. 23rd Symp. on the Interface Interface Foundation, Fairfax Station: 156–63.

58 Goldman, N., J. P. Anderson and A. G. Rodrigo. 2000. Likelihood-based tests of topologies in phylogenetics. Syst. Biol. **49**: 652–70.

59 Goldman, N. and Z. Yang. 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. Mol. Biol. Evol. **11**: 725–36.

60 Gordon, A. D. 1986. Consensus supertrees: the synthesis of rooted trees containing overlapping sets of labelled leaves. J. Classif. **3**: 335–48.

61 Gu, X., Y.-X. Fu and W.-H. Li. 1995. Maximum likelihood estimation of the heterogeneity of substitution rate among nucleotide sites. Mol. Biol. Evol. **12**: 546–57.

62 Guindon, S. and G. Perrière. 2005. Intragenomic base content variation is a potential source of biases when searching for horizontally transferred genes. Mol. Biol. Evol. **18**: 1838–40.

63 Guindon, S. and O. Gascuel. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst. Biol. **52**: 696–704.

64 Gusfield, D. and V. Bansal. 2005. A fundamental decomposition theory for phylogenetic networks and incompatible characters. Proc. RECOMB **9**: 217–32.

65 Hasegawa, M., H. Kishino and T.-A. Yano. 1985. Dating of the human–ape splitting by a molecular clock of mitochondrial DNA. J. Mol. Evol. **22**: 160–74.

66 Hasegawa, M. and H. Kishino. 1994. Accuracies of the simple methods for estimating the bootstrap propability of a maximum-likelihood tree. Mol. Biol. Evol. **11**: 142–5.

67 Hendy, M. D. and D. Penny. 1982. Branch and bound algorithms to determine minimal evolutionary trees. Math. Biosci. **60**: 133–42.

68 Hennig, W. 1966. *Phylogenetic systematics.* University of Illinois Press, Urbana, IL.

69 Hillis, D. M. 1996. Inferring complex phylogenies. Nature **383**: 130–1.

70 Holland, B. R., D. Penny and M. D. Hendy. 2003. Outgroup misplacement and phylogenetic inaccuracy under a molecular clock – a simulation study. Syst. Biol. **52**: 229–38.

71 Holliday, R. and G. W. Grigg. 1993. DNA methylation and mutation. Mutat. Res. **285**: 61–7.

72 Huber, K. T., M. Langton, D. Penny, V. Moulton and M. Hendy. 2002. Spectronet: a package for computing spectra and median networks. Appl. Bioinform. **20**: 159–61.

73 Huber, K. T. and V. Moulton. 2005. Phylogenetic Networks. In Gascuel. (ed.), *Mathematics of Evolution and Phylogeny*. Oxford University Press, Oxford: 178–204.

74 Huelsenbeck, J. P., B. Larget and M. E. Alfaro. 2004. Bayesian phylogenetic model selection using reversible jump Markov chain Monte Carlo. Mol. Biol. Evol. **21**: 1123–33.

75 Huelsenbeck, J. P., B. Larget, R. Miller and F. Ronquist. 2002. Potential applications and pitfalls of Bayesian inference of phylogeny. Syst. Biol. **51**: 673–88.

76 Huelsenbeck, J. P., F. Ronquist, R. Nielsen and J. P. Bollback. 2001. Bayesian inference of phylogeny and its impact on evolutionary biology. Science **294**: 2310–4.

77 Huelsenbeck, J. P. and F. Ronquist. 2001. MRBAYES: Bayesian inference of phylogenetic trees. Bioinformatics **17**: 754–5.

78 Huson, D. H., T. Dezulian, T. Klöpper and M. A. Steel. 2004. Phylogenetic super-networks from partial trees. In Proc. 4th Workshop on Algorithms in Bioinformatics, Bergen: 388–99.

79 Huson, D. H., T. Kloepper, P. J. Lockhart and M. Steel. 2005. Reconstruction of reticulate networks from gene trees. Proc. RECOMB **9**: 233–49.

80 Huson, D. H. 1998. SplitsTree: analyzing and visualizing evolutionary data. Bioinformatics **14**: 68–73.

81 Jermiin, L. S., G. J. Olsen, K. L. Mengersen and S. Easteal. 1997. Majority-rule consensus of phylogenetic trees obtained by maximum-likelihood analysis. Mol. Biol. Evol. **14**: 1296–302.

82 Jukes, T. H. and C. R. Cantor. 1969. Evolution of protein molecules. In Munro, H. N. (ed.), *Mammalian Protein Metabolism*. Academic Press, New York: 21–123.

83 Khaitovich, P., S. Pääbo and G. Weiss. 2005. Toward a neutral evolutionary model of gene expression. Genetics **170**: 929–39.

84 KIMURA, M. 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. J. Mol. Evol. **16**: 111–20.

85 KING, M.-C. AND A. C. WILSON. 1975. Evolution at two levels in humans and chimpanzees. Science **188**: 107–16.

86 KLEMPA, B., H. A. SCHMIDT, R. ULRICH, S. KALUZ, M. LABUDA, H. MEISEL, B. HJELLE AND D. H. KRÜGER. 2003. Genetic interaction between distinct Dobrava Hantavirus subtypes in *Apodemus agrarius* and *A. flavicollis* in nature. J. Virol. **77**: 804–9.

87 KLOTZ, L. C., N. KOMAR, R. L. BLANKEN AND R. M. MITCHELL. 1979. Calculation of evolutionary trees from sequence data. Proc. Natl. Acad. Sci. USA **76**: 4516–20.

88 KLUGE, A. G. AND J. S. FARRIS. 1969. Quantitative phyletics and the evolution of anurans. Syst. Zool. **18**: 1–32.

89 KLUGE, A. G. 1989. A concern for evidence and a phylogenetic hypothesis of relationships among Epicrates (Boidae, Serpentes). Syst. Zool. **38**: 7–25.

90 KLUGE, A. G. 1998. Total evidence or taxonomic congruence: cladistics or consensus classification. Cladistics **14**: 151–8.

91 KURLAND, C. G., B. CANBACK AND O. G. BERG. 2005. Horizontal gene transfer: a critical view. Proc. Natl. Acad. Sci. USA **100**: 9658–62.

92 LANAVE, C., G. PREPARATA, C. SACCONE AND G. SERIO. 1984. A new method for calculating evolutionary substitution rates. J. Mol. Evol. **20**: 86–93.

93 LAND, A. AND A. DOIG. 1960. An automatic method for solving discrete programming problems. Econometrica **28**: 497–520.

94 LAPOINTE, F.-J. AND G. CUCUMEL. 1997. The average consensus procedure: combining of weighted trees containing identical or overlapping sets of taxa. Syst. Biol. **46**: 306–12.

95 LARGET, B. AND D. L. SIMON. 1999. Markov chain Monte Carlo algorithms for Bayesian analysis of phylogenetic trees. Mol. Biol. Evol. **16**: 750–9.

96 LEMMON, A. R. AND M. C. MILINKOVITCH. 2002. The metapopulation genetic algorithm: an efficient solution for the problem of large phylogeny estimation. Proc. Natl. Acad. Sci. USA **99**: 10516–21.

97 LIÒ, P. AND N. GOLDMAN. 1998. Models of molecular evolution and phylogeny. Genome Res. **8**: 1233–44.

98 LOCKHART, P. J., P. A. MCLENACHAN, D. HAVELL, D. GLENNY, D. HUSON AND U. JENSEN. 2001. Phylogeny, radiation, and transoceanic dispersal of New Zealand alpine buttercups: molecular evidence under split decomposition. Ann. Missouri Bot. Gard. **88**: 458–77.

99 MADDISON, D. R., M. RUVOLO AND D. L. SWOFFORD. 1992. Geographic origins of human mitochondrial DNA: phylogenetic evidence from control region sequences. Syst. Biol. **41**: 111–24.

100 MADDISON, W. P., M. J. DONOGHUE AND D. R. MADDISON. 2005. Outgroup analysis and parsimony. Syst. Zool. **33**: 83–103.

101 MAKARENKO, V. 2001. T-REX: reconstructing and visualizing phylogenetic trees and reticulation networks. Bioinformatics **17**: 664–8.

102 MARGUSH, T. AND F. R. MCMORRIS. 1981. Consensus *n*-trees. Bull. Math. Biol. **43**: 239–44.

103 MARTIN, W. 1999. Mosaic bacterial chromosomes: a challenge en route to a tree of genomes. BioEssays **21**: 99–104.

104 MASSINGHAM, T. AND N. GOLDMAN. 2005. Detecting amino acid sites under positive selection and purifying selection. Genetics **169**: 1753–62.

105 MAU, B., M. A. NEWTON AND B. LARGET. 1999. Bayesian phylogenetic inference via Markov chain Monte Carlo methods. Biometrics **55**: 1–12.

106 MILLER, R. G. 1974. The Jackknife – a review. Biometrika **61**: 1–15.

107 MINH, B. Q., L. S. VINH, A. VON HAESELER AND H. A. SCHMIDT. 2005. pIQPNNI – parallel reconstruction of large maximum likelihood phylogenies. Bioinformatics **21**: 3794–6.

108 MIYAMOTO, M. M. AND W. M. FITCH. 1995. Testing species phylogenies and phylogenetic methods with congruence. Syst. Biol. **44**: 64–76.

109 Nieselt-Struwe, K. and A. von Haeseler. 2001. Quartet-mapping, a generalization of the likelihood-mapping procedure. Mol. Biol. Evol. **18**: 1204–19.

110 Nixon, K. C. and J. M. Carpenter. 1993. On outgroups. Cladistics **9**: 413–26.

111 Olsen, G. J., H. Matsuda, R. Hagstrom and R. Overbeek. 1994. fastDNAml: a tool for construction of phylogenetic trees of DNA sequences using maximum likelihood. Comput. Appl. Biosci. **10**: 41–8.

112 Page, R. D. M. 2002. Modified Mincut supertrees. In Proc. 2nd Workshop on Algorithms in Bioinformatics, Rome: 537–51.

113 Paraskevis, D., K. Deforche, P. Lemey, G. Magiorkinis, A. Hatzakis and A.-M. Vandamme. 2005. SlidingBayes: exploring recombination using a sliding window approach based on Bayesian phylogenetic inference. Bioinformatics **21**: 1274–5.

114 Patterson, C. 1988. Homology in classical and molecular biology. Mol. Biol. Evol. **5**: 603–25.

115 Piaggio-Talice, R., G. Burleigh and O. Eulenstein. 2004. Quartet supertrees. In Bininda-Emonds, O. R. P. (ed.), *Phylogenetic Supertrees: Combining Information to Reveal the Tree of Life*. Kluwer, Dordrecht: 173–91.

116 Posada, D. and K. A. Crandall. 2001. Intraspecific gene genealogies: trees grafting into networks. TREE **16**: 37–45.

117 Posada, D. and K. A. Crandall. 2002. The effect of recombination on the accuracy of phylogeny estimation. J. Mol. Evol. **54**: 396–402.

118 Purvis, A. 1995. A composite estimate of primate phylogeny. Philos. Trans. R. Soc. Lond. B **348**: 405–21.

119 Ragan, M. A. 1992. Phylogenetic inference based on matrix representation of trees. Mol. Phylogenet. Evol. **1**: 53–8.

120 Roberts, D. L. and A. R. Solow. 2003. Flightless birds: when did the dodo become extinct? Nature **426**: 245.

121 Robinson-Rechavi, M. and D. Graur. 2001. Usage optimization of unevenly sampled data through the combination of quartet trees: an eutherian draft phylogeny based on 640 nuclear and mitochondrial proteins. Isr. J. Zool. **47**: 259–70.

122 Rohlf, F. J. 1982. Consensus indices for comparing classifications. Math. Biosci. **59**: 131–44.

123 Saitou, N. and M. Nei. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. **4**: 406–25.

124 Salemi, M. and A.-M. Vandamme. 2003. *The Phylogenetic Handbook: A Practical Approach to DNA and Protein Phylogeny*. Cambridge University Press, Cambridge.

125 Salminen, M. O., J. K. Carr, D. S. Burke and F. E. McCutchan. 1995. Genotyping of HIV-1. *Human Retroviruses and AIDS Compendium.* Los Alamos National Laboratory III-30–III-34, Los Alamos, NM.

126 Sanderson, M. J., A. Purvis and C. Henze. 1998. Phylogenetic supertrees: assembling the trees of life. TREE **13**: 105–9.

127 Sanderson, M. J. and H. B. Shaffer. 2002. Troubleshooting molecular phylogenetic analyses. Annu. Rev. Ecol. Syst. **33**: 49–72.

128 Sankoff, D. and J. B. Kruskal. 1983. Time warps, string edits, and macromolecules. Addison-Wesley, Reading, MA.

129 Sankoff, D. 1975. Minimal mutation trees of sequences. SIAM J. Appl. Math. **28**: 35–42.

130 Satta, Y., J. Klein and N. Takahata. 2000. DNA archives and our nearest relative: the trichotomy problem revisited. Mol. Phylogenet. Evol. **32**: 259–75.

131 Sattath, S. and A. Tversky. 1977. Additive similarity trees. Psychometrika **42**: 319–45.

132 Schmidt, H. A., K. Strimmer, M. Vingron and A. von Haeseler. 2002. TREE-PUZZLE: maximum likelihood phylogenetic analysis using quartets and parallel computing. Bioinformatics **18**: 502–4.

133 Schmidt, H. A. 2003. *Phylogenetic Trees from Large Datasets.* PhD Thesis. Universität Düsseldorf.

134 Schuh, R. T. 2000. *Biological Systematics: Principles and Applications.* Cornell University Press, Ithaca, NY.

135 Semple, C. and M. Steel. 2000. A supertree method for rooted trees. Discr. Appl. Math. **105**: 147–58.

136 Semple, C. and M. Steel. 2003. *Phylogenetics.* Oxford University Press, Oxford.

137 Sheldon, F. H. 1987. Phylogeny of herons estimated from DNA–DNA hybridization data. Auk **104**: 97–108.

138 Shimodaira, H. and M. Hasegawa. 1999. Multiple comparisons of log-likelihoods with applications to phylogenetic inference. Mol. Biol. Evol. **16**: 1114–6.

139 Sinsheimer, J. S., J. A. Lake and R. J. A. Little. 1996. Bayesian hypothesis testing of four-taxon topologies using molecular sequence data. Biometrics **52**: 715–28.

140 Smith, A. B. 1994. Rooting molecular trees: problems and strategies. Biol. J. Linn. Soc. **51**: 279–92.

141 Stamatakis, A. P., T. Ludwig and H. Meier. 2005. RAxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees. Bioinformatics **21**: 456–63.

142 Stewart, C. A., D. Hart, D. K. Berry, G. J. Olsen, E. A. Wernert and W. Fischer. 2001. Parallel implementation and performance of fastDNAml – a program for maximum likelihood phylogenetic inference. In Proc. Int. Conf. on High Performance Computing and Communications, Denver, CO: 191–201.

143 Strimmer, K., C. Wiuf and V. Moulton. 2001. Recombination analysis using directed graphical models. Mol. Biol. Evol. **18**: 97–9.

144 Strimmer, K. and A. von Haeseler. 1996. Quartet puzzling: a quartet maximum-likelihood method for reconstructing tree topologies. Mol. Biol. Evol. **13**: 964–9.

145 Strimmer, K. and A. von Haeseler. 1997. Likelihood-mapping: a simple method to visualize phylogenetic content of a sequence alignment. Proc. Natl Acad. Sci. USA **94**: 6815–9.

146 Strimmer, K. and V. Moulton. 2000. Likelihood analysis of phylogenetic networks using directed graphical models. Mol. Biol. Evol. **17**: 875–81.

147 Studier, J. A. and K. J. Keppler. 1988. A note on the neighbor-joining algorithm of Saitou and Nei. Mol. Biol. Evol. **5**: 729–31.

148 Swofford, D. L., G. J. Olsen, P. J. Waddell and D. M. Hillis. 1996. Phylogeny reconstruction. In Hillis, D. M., C. Moritz, and B. K. Mable (eds.), *Molecular Systematics*, Sinauer, Sunderland, MA: 407–514.

149 Tamura, K. and M. Nei. 1993. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. Mol. Biol. Evol. **10**: 512–26.

150 Tarrío, R., F. Rodríguez-Trelles and F. J. Ayala. 2001. Shared nucleotide composition biases among species and their impact on phylogenetic reconstructions of the drosophilidae. Mol. Biol. Evol. **18**: 1464–73.

151 Vinh, L. S., H. A. Schmidt and A. von Haeseler. 2005. PhyNav: a novel approach to reconstruct large phylogenies. In Weihs, C. and W. Gaul (eds.), *Classification – The Ubiquitous Challenge*. Springer, Berlin: 386–93.

152 Vinh, L. S. and A. von Haeseler. 2004. IQPNNI: moving fast through tree space and stopping in time. Mol. Biol. Evol. **21**: 1565–71.

153 von Haeseler, A. and G. A. Churchill. 1993. Network models for sequence evolution. J. Mol. Evol. **37**: 77–85.

154 Wheeler, W. C. 2005. Nucleic acid sequence phylogeny and random outgroups. Cladistics **6**: 363–8.

155 Wilkinson, M., J. L. Thorley, D. Pisani, F.-J. Lapointe and J. O. McInerney. 2004. Some desiderata for liberal supertrees. In Bininda-Emonds, O. R. P. (ed.), *Phylogenetic*

Supertrees: Combining Information to Reveal the Tree of Life. Kluwer, Dordrecht: 227–46.

156 YANG, Z. AND R. NIELSEN. 1998. Synonymous and nonsynonymous rate variation in nuclear genes of mammals. J. Mol. Evol. **46**: 409–18.

157 YANG, Z. AND W. J. SWANSON. 2002. Codon-substitution models to detect adaptive evolution that account for heterogeneous selective pressures among site classes. Mol. Biol. Evol. **19**: 49–57.

158 ZILLIG, W. 1991. Comparative biochemistry of Archaea and Bacteria. Curr. Opin. Genet. Dev. **1**: 544–51.

5
Finding Protein-coding Genes

David C. Kulp

1 Introduction

Gene finding in genomic DNA sequences is a critical step in the functional annotation of genomes. Over the past approximately quarter century increasingly sophisticated methods have been developed to better understand and catalog the mechanisms of transcription, splicing and translation, and to predict the gene products, be they peptide sequences or RNA genes. With the advent of large-scale sequencing, software programs were developed to automate gene prediction.

In this chapter the common techniques for computational gene finding are introduced. Basic concepts and terminology are given in Section 2. Sections 3–5 discuss feature prediction for both content and signal features, and Section 6 introduces the standard dynamic programming formalism for incorporating multiple features into complete gene structure predictions. Some performance results for *ab initio* gene finding are given in Section 7. Practical gene finding must also consider available experimental mRNA, protein and genomic sequence data. Some of these homology methods as well as other integrative approaches are described in Section 8. Finally, the chapter concludes with some caveats about the practical limitations of automated gene prediction.

2 Basic DNA Terminology

Since one DNA strand is the complement of the other, in DNA analysis only one strand is stored in the databases. Which strand is represented is generally arbitrary and unimportant, but in this chapter the represented sequence is called the forward strand and the implicit complement is the reverse. A DNA sequence is always represented, by convention, in the direction of DNA replication. The left end of the sequence is referred to as upstream or 5′ and the right end is downstream or 3′.

Bioinformatics - From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

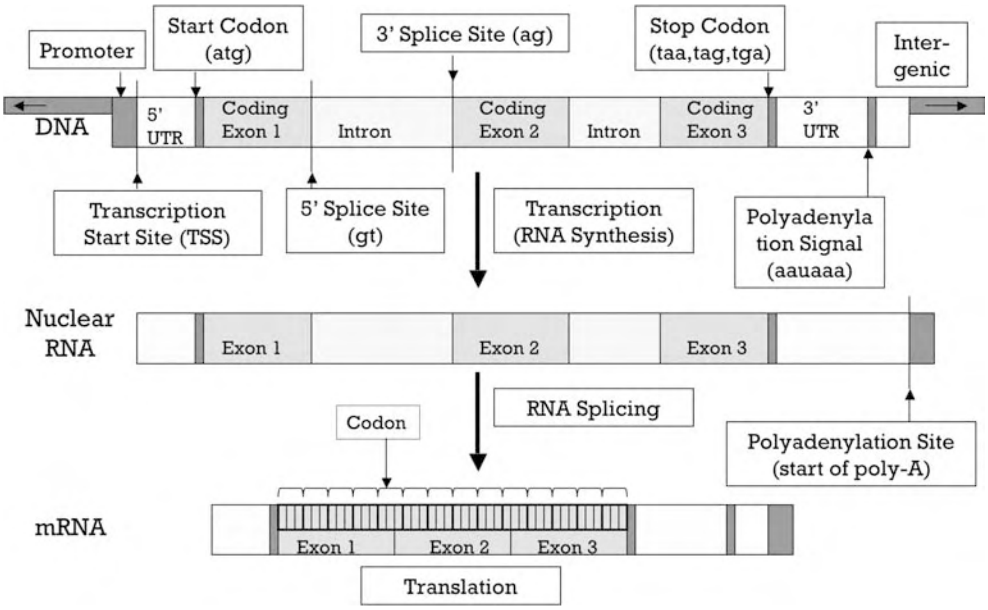


Figure 1 The central dogma of molecular biology in eukaryotes. A primary transcript region starting at the promoter is copied into pre-mRNA. The transcript is then spliced in eukaryotes to produce the mature cytoplasmic message. The message is translated into peptides. Note that codons may span splice boundaries. Although not shown in this diagram, splicing is possible in the untranslated regions.

For the purposes of this discussion, a gene is defined as the subsequence of genomic DNA that is transcribed by RNA polymerase – usually Pol II when referring to eukaryotic transcription. The gene structure further includes those features on the mRNA involved with splicing and translation, i.e. the splice and translation start and stop sites. For convenience, all of these features are usually annotated with respect to the original DNA sequence as shown in Figure 1. Transcription occurs on single-stranded DNA, on either the forward or reverse strand. By convention the gene structure is annotated on the informational or sense strand (not the template or antisense strand).

Although transcription has been observed to occur at the same physical genomic position on both strands, we usually assume for simplicity that there are no overlapping transcripts. Automatic gene finders must evaluate both the explicitly represented sequence and the implicit reverse complement – this is usually performed simultaneously.

Predicting genes in eukaryotes is considerably more challenging than in prokaryotes because of splicing. Most transcribed mRNA (pre-mRNA) in eukaryotes is spliced into smaller sequences called processed or mature mRNA through the excision of introns by the spliceosome complex, leaving a set of concatenated exons to be passed to the ribosome. The introns are located between the 5′ and 3′ splice sites, also called donor and acceptor sites. At least 99% of 5′ splice sites begin with "GT" and 3′ splice sites end with "AG", called the consensus dinucleotides.

The spliceosome concatenates exons separated by often long introns. Each exon can be as short as a few nucleotides. Thus, while gene finding in prokaryotes involves indentifying a single contiguous coding sequence, gene finding in eukaryotes requires a combinatorial search of many different possible exons.

3 Detecting Coding Sequences

Identification of the protein-coding domain sequence (CDS) between the start and stop codons is of great interest because the translated peptide product can be directly inferred from the CDS. The ribosome, after binding to the mRNA, begins translation at an AUG triplet (ATG in DNA). The ribosome matches these triplets, called codons, consecutively with tRNAs adding deterministically one of the 20 amino acids to a polypeptide sequence for each codon according to the genetic code (Table 1). The translation process is terminated when one of three stop codons (UAA, UAG or UGA) is encountered. Thus, the CDS on the DNA sequence is composed of a sequence of codons that code for

Table 1 The standard genetic code showing codon and amino acid single- and three-letter abbreviations

TTT	F	Phe	TCT	S	Ser	TAT	Y	Tyr	TGT	C	Cys
TTC	F	Phe	TCC	S	Ser	TAC	Y	Tyr	TGC	C	Cys
TTA	L	Leu	TCA	S	Ser	TAA	*	Ter	TGA	*	Ter
TTG	L	Leu	TCG	S	Ser	TAG	*	Ter	TGG	W	Trp
CTT	L	Leu	CCT	P	Pro	CAT	H	His	CGT	R	Arg
CTC	L	Leu	CCC	P	Pro	CAC	H	His	CGC	R	Arg
CTA	L	Leu	CCA	P	Pro	CAA	Q	Gln	CGA	R	Arg
CTG	L	Leu	CCG	P	Pro	CAG	Q	Gln	CGG	R	Arg
ATT	I	Ile	ACT	T	Thr	AAT	N	Asn	AGT	S	Ser
ATC	I	Ile	ACC	T	Thr	AAC	N	Asn	AGC	S	Ser
ATA	I	Ile	ACA	T	Thr	AAA	K	Lys	AGA	R	Arg
ATG	M	Met	ACG	T	Thr	AAG	K	Lys	AGG	R	Arg
GTT	V	Val	GCT	A	Ala	GAT	D	Asp	GGT	G	Gly
GTC	V	Val	GCC	A	Ala	GAC	D	Asp	GGC	G	Gly
GTA	V	Val	GCA	A	Ala	GAA	E	Glu	GGA	G	Gly
GTG	V	Val	GCG	A	Ala	GAG	E	Glu	GGG	G	Gly

Prokaryotes also use an additional GTG initiation codon. There are other rare genetic codes as well. See http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi.

the corresponding protein, beginning with the start codon and ending with a stop codon.

Note that the notions CDS and exon are not synonymous, although frequently exchanged in gene-finding literature. Exons refer to those DNA segments that are not excised by splicing, i.e. all of the sequence corresponding to the mature mRNA. Exonic sequences can be either coding (CDS) or untranslated regions (UTRs). A CDS typically begins and ends in the middle of an exon. Introns are possible upstream and (rarely) downstream of a CDS. Splicing need not occur on codon boundaries.

3.1 Reading Frames

There are six possible reading frames along double-stranded DNA – three on each strand. A CDS beginning translation at position i is in reading frame $f = i$ modulo 3. Reading frames on the opposite strand are conventionally labeled as $-f$. We say that a codon is in frame if its position modulo 3 is the same as the CDS in question. In particular, an in-frame stop codon terminates a CDS, but out-of-frame stop codons have no effect. A sequence of consecutive codons between a start and stop codon is called an open reading frame (ORF).

Genes in prokaryotes are relatively easy to identify by searching for ORFs of a minimum length, say 300 nucleotides. In random DNA, an in-frame stop codon is expected about every 21 codons (63 nucleotides) and the chance of an ORF longer than this becomes increasingly unlikely. Small ORFs can truly be coding, especially in eukaryotes, due to splicing and asymmetric nucleotide distributions can easily allow for long ORFs, requiring more sophisticated pattern recognition methods, as described below.

3.2 Coding Potential

The distribution of codons is subject to evolutionary and biophysical constraints. The G + C content (fraction of G and C nucleotides) among genomes varies, which affects codon frequencies for different organisms. Amino acid frequencies are not uniform and arrangements of amino acids in polypeptides are, of course, also not random. These effects, as well as other DNA and mRNA structural and processing constraints, lead to biases in the frequency and ordering of codons, called codon bias. Moreover, basal expression levels have been observed to relate to the levels of available tRNAs, so codons in higher expressed genes are more significantly biased towards the abundant tRNAs [27]. Synonymous codon bias, closely related to codon bias, describes the differing frequencies of codons coding for the same amino acid.

Codon usage tables that list codon frequencies have been compiled for many organisms (http://www.kazusa.or.jp/codon). To test for coding potential is to assess whether the frequencies of codons in a candidate ORF are statistically similar to the codon usage for the organism and these measures are typically the heart of all gene-finding programs. For example, about 2.1% of human codons are "ATC" and among its class of synonymous codons that encode isoleucine ("ATC", "ATT" and "ATA"), "ATC" is used about 47% of the time.

A representative method of this class of coding potential measures is the Gribskov codon preference statistic [26]. The relative frequency of a codon, C, among its class of synonymous codons, $f_{\text{class}(C)}(C)$, is computed using a codon-usage table derived from highly expressed genes and compared to the relative frequency of the codon, $g_{\text{class}(C)}(C)$, in a background model according to position-independent nucleotide composition:

$$S(C) = \log\left[f_{\text{class}(C)}(C)/g_{\text{class}(C)}(C)\right] \ .$$

For example, suppose the codon, C, is "ATC", and its codon usage is as above and the frequencies of A, T, C and G in the query sequence are (0.21, 0.21, 0.29, 0.29). Then $S(C = \text{ATC}) = \log(0.47/0.41) = 0.14$. The normalized sum of log-likelihood ratios over a window w (of, say, 25 codons) provides an indication of relative coding potential and expression. By convention, the normalized sum is exponentiated to generate the final Gribskov statistic. Such methods are often used in plots for preliminary visualization of a novel genomic sequence and are sometimes sufficient to support manual gene prediction in prokaryotes (Figure 2).

Measures such as the Gribskov codon preference statistic lack consideration for the positions of codons relative to each other. Observed dependencies among adjacent codons led to the proposal of several measures based on pairs of codons (dicodons). In an important benchmark paper, Fickett and Tung [22], assessed most of the extant methods and their conclusion was that dicodon (or hexamer, more generally) measurements were superior to all other methods.

The three-periodic fifth-order Markov model is a particularly appealing formulation of hexamer statistics that is widely used in modern gene finders. Proposed by Borodovsky [7], such Markov models are used to represent the probability distributions of the four possible nucleotides at each of the three base positions in a codon. Suppose we are interested in a codon beginning at position i composed of individual nucleotides b_i, b_{i+1} and b_{i+2}. Three separate Markov models are defined: $P_0(b_i|b_{i-5}\dots b_{i-1})$, $P_1(b_{i+1}|b_{i-4}\dots b_i)$ and $P_2(b_{i+2}|b_{i-3}\dots b_{i+1})$. Each probability distribution is generated from simple frequency counts of each possible nucleotide in the context of the previous five nucleotides. Training sets of millions of codons are available from annotated

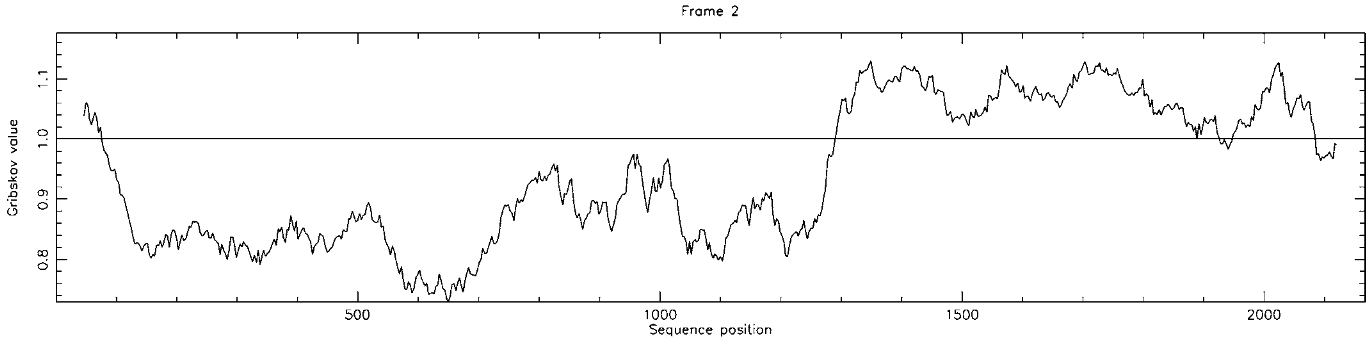


Figure 2 Codon bias computed by the Gribskov codon preference statistic. Frame 2 of the *Pseudomonas amiC* gene as generated by the program syco from the EMBOSS software collection. The region starting around position 1300 has consistently high coding potential, suggesting a coding region.

GenBank sequences. Assuming conditional independence, the probability of a codon is just the product of the probabilities of the individual bases b_i, b_{i+1}, and b_{i+2}. A separate null model is defined for noncoding bases, $P_{\mathrm{nc}}(b_i|b_{i-5}\dots b_{i-1})$. A log-likelihood ratio score for a codon starting at i is then:

$$\sum_{f=0..2}\left[\log P_f(b_{i+f}|b_{i+f-5}\dots b_{i+f-1}) - \log P_{\mathrm{nc}}(b_{i+f}|b_{i+f-5}\dots b_{i+f-1})\right] .$$

These scores can be accumulated over a window as with the Gribskov measure.

To obtain accurate estimates, a fifth-order Markov model requires sufficiently large numbers of observations in each context. This is not always available and, in some cases, even longer contexts may be plentiful. Salzberg describes a variant that uses different context length depending on the available data [66].

The fact that statistics related to codon usage aid in the identification of CDS and also in the estimation of expression levels indicates an inherent classification weakness in the use of codon statistics, i.e. that genes with low expression levels are more difficult to find because their statistics are weaker. Low expressors are more difficult to detect experimentally as well, further biasing the codon statistics gathered regarding known genes.

4 Gene Contents

In addition to the statistical regularities in CDS, other discriminating properties of coding exons and introns have been observed. These features of variable-length DNA sequences are sometimes referred to as content.

For example, noncoding DNA is expected to have a relatively neutral distribution of nucleotides with exceptions such as physical–chemical constraints and the presence of repeats. Thus, models of noncoding DNA have been devised similar to coding potential measures. A simple and common implementation is the use of Markov models for intron and intergenic DNA analagous to the three fifth-order Markov models for coding DNA. The log probabilities of different models can then be compared. Figure 3 shows the distribution of probabilities from fifth-order coding and intron models, and the distribution of the difference in log probabilities.

Guigo and Fickett [29] showed that all content measures are highly correlated with G + C bias. Thus, it is common to compute Markov distributions by partitioning the training data into discrete isochores (extended regions of G + C bias in the genome) based on windowed-G + C content [9].

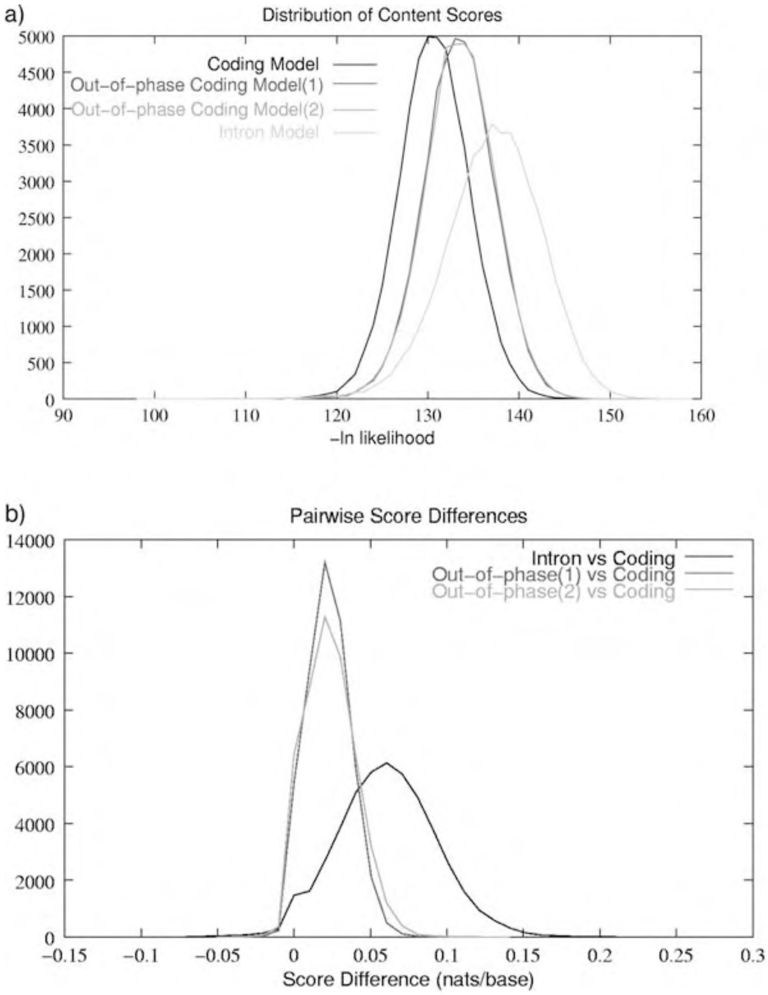


Figure 3 Score distributions for Markov chain models of coding and intronic DNA. Three Markov chains were trained on 53 183 460 coding bases and one Markov chain was trained on 16 149 264 intronic bases from the well-annotated protein-coding exons of *Caenorhabdtis elegans*. For each of 48 124 exons, a 99-nucleotide in-frame coding region was scored using the coding model, two out-of-phase coding models and the intron model. (a) The distribution of log-likelihoods for fifth-order Markov chains. (b) The distribution of differences in log-likelihoods per base of the fifth-order coding model versus the intron and out-of-phase fifth-order models for each coding exon. (Pairwise comparisons with likelihoods from out-of-phase models were only made if there were no in-frame stop codons in the alternative frame.) The implication here is that although the overall distributions for the coding and noncoding models are very similar, a comparison of scores for individual exons shows good separation (i.e. most model score differences are greater than zero). This simply shows that the fifth-order Markov models are reasonably good at classifying coding regions.

The lengths of exons and introns differ – often significantly [32]. Exons follow an approximately log-normal distribution with a mean length of about 140 bases in most eukaryotes, but the typical length of introns varies significantly by organism. Many of the model organisms such as fly and worm have intron lengths within a relatively tight range of about 70 bases – the minimal required intron length for efficient splicing. Mammalian introns are typically much longer than exons due to prolific insertions of repetitive elements; they are rarely less than 100 bases and have a long exponential distribution to 10^6 bases.

5 Gene Signals

Gene structure is defined by the start and stop positions in DNA of exons and CDS. Through laboratory experimentation, molecular biologists have shown that for each of these sites there are necessary, conserved motifs that govern the transcription and translational machinery. With respect to gene finding, Staden [75] distinguished these control sites as signals as compared to variable length content. Signal features loosely correspond to binding sites or special functional patterns recognized by the polymerase, spliceosome or ribosome.

If it were possible to automatically detect all signals independently, then the gene-finding solution would be complete. For the most part, however, no one signal can reliably be detected on its own. Later in this chapter we will learn how to combine these measures along with coding potential to achieve superior gene finding performance. First, we explore a few of the methods for independent signal detection.

5.1 Splice Sites

Degenerate matches to a motif can be detected using a position-specific weight matrix [74]. Weight matrices are commonly used in many biosequence approximate matching problems. A weight matrix is a (2-D) array $W(1 \leq i \leq m, 1 \leq j \leq 4)$ such that $W(i, j)$ is the probability of nucleotide j at position i in a motif of length m. Frequencies can be used to generate these probabilities, priors can be introduced when data is sparse or more sophisticated contexts can be represented such as dinucleotide frequencies (e.g. an order-1 Markov weight matrix at each position resulting in a $m \times 16$ matrix). For example, almost all introns begin with the consensus dinucleotide “GT” and end with “AG”, but the regions around the beginning and end of the intron (the splice sites) have less specific nucleotide patterns. Figure 4 shows examples of weight matrices for splice sites in *C. elegans*.

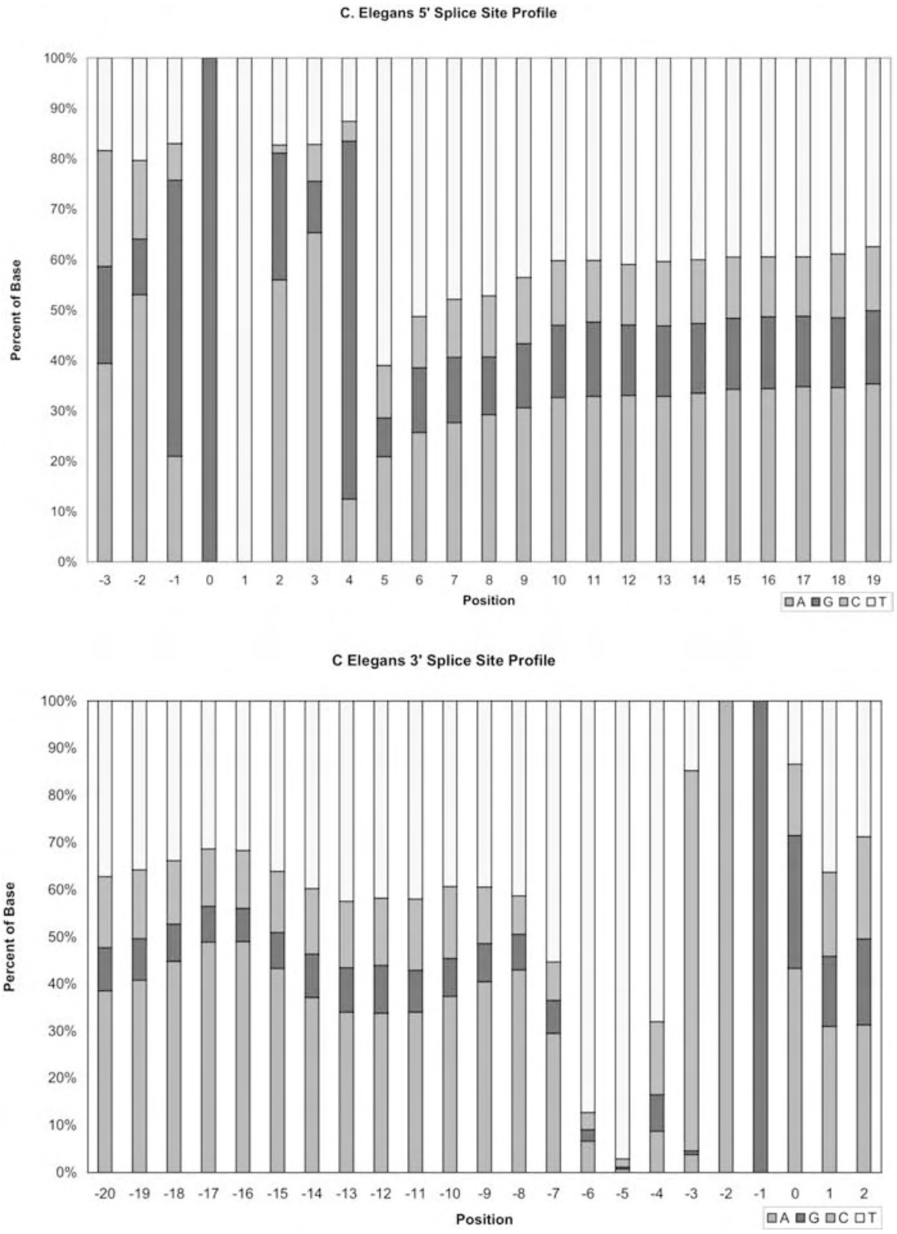


Figure 4 *C. elegans* splice site weight matrices. Windows of 20 bases downstream of the $5'$ exon junction ("GT", the beginning of the intron, is positions 0 and 1) and 20 bases upstream of the $3'$ exon junction ("AG", the end of the intron, is positions -1 and -2) were selected from curated gene sequences [16].

Given any test sequence represented as a 2-D matrix $S(1 \le i \le m, 1 \le j \le 4)$, where $S_{i,j} = 1$ for the nucleotide *j* found at position *i* and 0 elsewhere, then the likelihood of the feature can be defined simple as $\prod_{i=1...m} \prod_{j=1...4} W_{i,j}^{S_{i,j}}$. Stormo presented thermodynamic, likelihood, and information theoretic justifications for the use of weight matrices [76].

Moving beyond the simple weight matrix is the maximal dependence decomposition decision tree (MDD) method that captures local, but nonadjacent dependencies [9]. The MDD method evaluates a set of rules to determine which weight matrix to use to score the sequence. The rules are based on the dependencies between positions in the motif. For example, in eukaryotic $5'$ splice sites, it can be shown that the distribution of nucleotides in the conserved columns $-3 \cdots + 6$ around the consensus GT are most correlated to the nucleotide G in position $+5$. Thus, the training set is partitioned according to the "+5" value into two sets such that the score function for each set is conditionally independent of that nonadjacent position. This leaves only adjacent dependencies or independent positions, which can be easily modeled using conventional weight matrices as above. (Other approaches have been proposed to detect dependencies among nonadjacent bases with similar performance characteristics, e.g. a Bayesian Network structure inference method was proposed by Cai and coworkers [13].)

The primary limitation of weight matrices is the inability to model insertions and deletions. To handle more complex motifs, profile hidden Markov models (HMMs) (see also Chapter 3) and related probabilistic state machine models can be employed in a similar manner as for protein sequences [20,42].

The $3'$ splice site is slightly more complicated because the upstream pyrimidine-rich branch site contributes to its recognition. However, the branch site is variable and so not amenible to fixed-width matrix methods. As a result, many recognition methods have been proposed for splice site recognition that allow for the incorporation of multiple distinct sequence features as inputs (branch site, splice site, intron content, exon content). One of the more effective techniques for combining different sequence features is discriminant analysis in which weights for different features are fitted to maximize the discrimination between true and decoy sites [72,82].

Probably the current leading method for splice site prediction is GeneSplicer – an extension of the MDD metric [59]. Other techniques include neural networks, boolean logic rules, decision trees, support vector machines (SVMs) and many others (e.g. Refs. [8,9,61,73]).

In addition to the conventional splice site recognition, about 1% of splice sites have non-canonical dinucleotides. This is largely ignored in gene finding, but is addressed by Burset and coworkers [11].

In general, recent methods achieve reasonable performance in splice site detection, but are unavoidably burdened by large numbers of decoy sites, resulting in false-positive rates around 5% when recognizing about 90% true splice sites.

5.2 Translation Initiation

Identifying the beginning of translation is perhaps surprisingly challenging. Part of the difficulty is that the database is rife with experimentally unconfirmed start sites. In addition, the signal for start sites tends to be rather weak. In prokaryotes, the Shine–Delgarno motif serves as a binding site for the ribosome preceding the first codon. The consensus motif is AGGAGG, but it can take on short and degenerate forms. Kozak observed that in higher eukaryotes translation usually begins at the first start codon after the transcription start site [41]. However, for the purposes of *ab initio* gene finding, this is usually of little help since the beginning of the transcript is also unknown and cannot be reliably predicted in large DNA sequences without a high false-positive rate [21] (and even the annotated transcription start sites are often wrong due to truncated mRNAs). In vertebrates, a consensus of `gccaccATGg` (start codon in caps) is observed and weight matrices have been developed from reliable start sites using first and second order models similar to approaches for splice sites [2, 40], but these methods are subject to high false-positive rates.

Like with splice site detection, many of the conventional machine learning techniques have been successfully applied including neural networks, linear and quadratic discriminant analysis and SVMs [58, 64, 67, 85]. The most successful independent predictor of translation initiation is an SVM classifier that remarkably identifies almost 100% of start sites with well less than 1% false positives on a standard test set [48].

5.3 Translation and Transcription Termination

Recognizing one of the stop codons (TAA, TAG or TGA) is trivial assuming that the reading frame is known. Conversely, more probable stops are those with high coding potential upstream of the site and low coding potential downstream. The transcript following the stop codon is typically one long exon. Splicing after the stop codon is rare.

Finally, transcription is terminated by a polyadenylation signal with a consensus of `AATAAA` although there are many variants. The motif is small, is not located predictably near other contextual features, is frequently unannotated in the databases and may not even be present. Moreover, it is estimated that in human about half of all transcripts have multiple $3'$ termination sites and these are often imprecisely cleaved [80]. Graber [25] describes a pseudo-probabilistic model for detecting termination in yeast. Again, standard weight matrix and discriminant analysis methods have shown moderate success for detecting the termination site [47,77]. In practice, transcription termination is largely ignored in *ab initio* prediction. Instead, it is often considered sufficient to detect just the gene structure from the start to stop codons.

6 Integrating Gene Features

So far, we have learned that there are different gene features (signals and contents) each with statistically significant discriminative power. There are numerous scoring methods for different features assessed independently, yet we intuit that combining these features is likely to yield better results. However, there is an exponential combination of possible labelings of exons, introns and intergenic regions (i.e. any segment can begin or end at any position). Thus, the gene finder is faced with two major problems: how to effectively combine features and how to efficiently explore the possible gene structures. The solution for both of these problems is using dynamic programming. In some implementations, logical adjacent features are combined into a single score and then a dynamic program is applied.

6.1 Combining Local Features

In the same way that multiple features were used as input to the long list of machine learning classifiers in splice site recognition (Section 5), so too can multiple features be combined to recognize larger functional units. Zhang is a major proponent of this strategy of recognizing exons based on combined information from the flanking signals and content, noting that the *in vivo* recognition of exons in transcription is believed to largely be driven by the interactions of DNA-binding complexes that straddle the exons, according to the exon definition model [5,30]. Zhang has produced a suite of methods for recognizing the $5'$ and $3'$ UTR exons, initial coding exon, internal coding exons and last coding exon using quadratic discriminant analysis, each recognition module combining multiple, different features, with excellent performance [77,83].

The choice of the fundamental functional units of gene recognition differ among gene finding programs. For example, the nucleotide is the basic unit in HMMGene, Genie treats splice sites and coding exons separately, and MZEF combines these local features into a single functional unit. However, in all cases, the same dynamic programming technique can be used to combine these functional units into complete gene structure predictions.

6.2 Dynamic Programming

Snyder and Stormo [71] showed how the optimal combination of features could be obtained using dynamic programming. Let us define the states of our gene finder as the different types of functional units in our gene model, $Q = q_1 \dots q_m$. We say that the sequence, $X = x_1 \dots x_n$ is labeled by a corresponding sequence of states, $\Phi = \phi_1 \dots \phi_n$, called the "parse", where $\phi_i \in Q$. Quite simply, a parse formally describes the gene structure, e.g. intergenic DNA from position 0 to 100, $5'$ UTR from 101 to 150, initial CDS exon from 151 to 200, etc.

The key idea behind dynamic programming is the assumption that the score of a parse can be decomposed into independent segments or, at least, segments that are only locally dependent. This independence assumption is clearly violated in some cases. For example, tertiary protein structure obviously implies specific long-range interactions among codons. Nevertheless, this is a reasonable approximation for gene finding that offers significant computational advantage.

If every possible segment can be scored independently, then the parse with the best score can be computed recursively. Given an input DNA sequence X and possible states Q, then define a score matrix $S(j,k)$ that holds the score of the best parse of the subsequence $x_1 \dots x_j$ ending with a segment at x_j in state q_k. Define $s(i,j,k)$ as the independent score of a segment from $x_i \dots x_j$ of state k. [These $s(i,j,k)$ terms are based on feature scores from one of the many methods discussed and alluded to in the previous sections, such as coding potential, splice site scores, etc.] If we assume for simplicity in this formulation that any segment of any length can follow any other segment (we will improve on this momentarily), then $S(j,k)$ is defined as the best score from all positions $i < j$ in any possible state in Q plus the score of a segment in state k from $i \dots j$. For example, the best score for labeling a DNA sequence such that that an initial exon ends at position 200 is computed by considering the score for an initial exon starting from every position less than 200 following any of the other possible states ($5'$ UTR, intron, intergenic, etc.):

$$S(j,k) = \max_{i<j, l \in Q} \left(S(i,l) + s(i,j,k) \right) . \tag{1}$$

The general form of this dynamic program is usually called the Viterbi algorithm and the process of predicting a parse is often called decoding [23]. While $S(\dot{;})$ holds only the best score, it is straightforward to also simultaneously compute a trace-back describing the parse that achieved the best score.

The formulation here is different from the conventional presentation of dynamic programming for biosequence analysis (Chapter 3) because segments can take on arbitrary length. As a result, running time is quadratic in the length of the sequence and number of states – prohibitively expensive except for very small sequences, but at least not exponential.

6.3 Gene Grammars

In order to ensure that the evaluation of all reasonable gene structures for long DNA sequences can be achieved in acceptable running time we add grammatical constraints that ensure only legal and sensible parses are considered. Dong and Searls [18, 68] were the first major proponents for describing gene structure in linguistic terms. (For a thorough treatment in a modern gene finding system, see also Ref. [46].) The key idea behind grammatical constraints is that different segment states can only appear within specific contexts. For example, an intron can only following an exon. The grammatical constraints for genes can be expressed in a so-called regular grammar and can be visualized as a finite state machine. Figure 5 shows such a state diagram for a simplified gene model.

We call neighboring pairs of states transitions (e.g. intron following exon). In addition to strict contextual constraints on state transitions, we observe that some state transitions are allowed, but are less likely than others. For example, it is less likely that an intron will be followed by a terminal exon than an internal exon. We define $t(l,k)$ as the score for a transition from state l to state k. These are usually assigned based on frequencies of observed transitions in training data. In addition, we define a function $T(k)$ that returns a set of allowable previous states for k.

One specific type of transition constraint in $T(\cdot)$ is especially important, i.e. the frame constraint. In order to ensure that the total number of bases in the CDS is a multiple of 3, states must be created to ensure that introns split codons in a frame consistent manner. For example, if one base precedes an intron then two bases must follow it before the next full codon (see Figure 5.)

In Section 4 we observed that exons and introns had predictable length distributions as well as maximum and minimum lengths that are rarely or never exceeded, e.g. coding exons are rarely larger than a few thousand bases and introns are almost never smaller than about 50 bases. Therefore, we restrict the allowable length segments considered in our dynamic program by introducing $\min(k)$ and $\max(k)$ values for each state k.

From this, we have an improved method for scoring possible parses that extends Eq. (1) as:

$$S(j,k) = \max_{\substack{i<j,\ l \in T(k) \\ j-i>\min(k) \\ j-i<\max(k)\ .}} S(i,l) + s(i,j,k) + t(l,k) \tag{2}$$

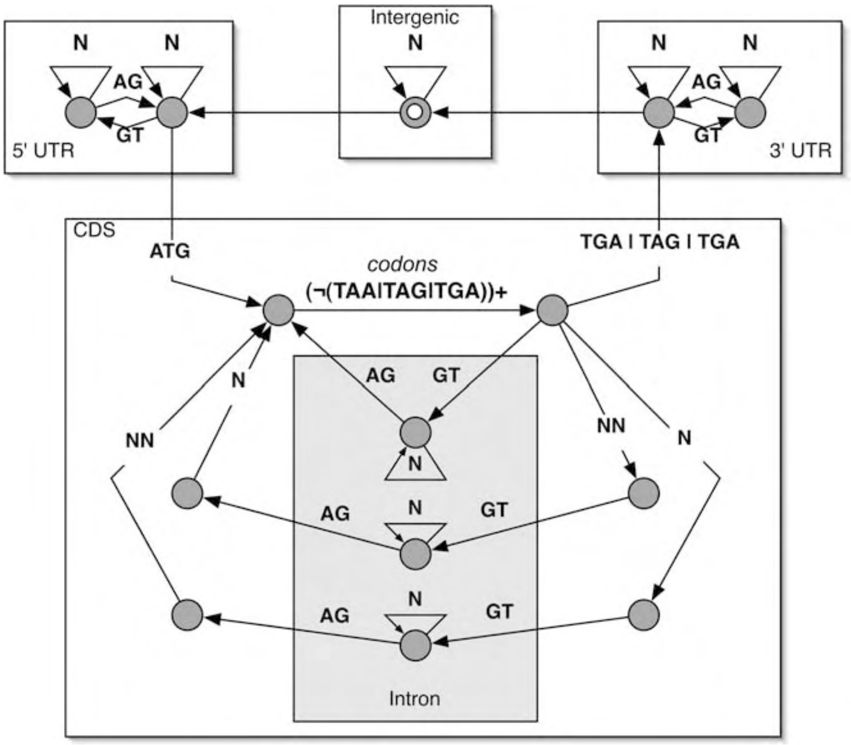


Figure 5 A finite state automaton (FSA) that recognizes gene structures. This simplified FSA recognizes legal protein-coding gene structures such that the total CDS length is a multiple of 3. Nucleotides are matched along the arcs and states are associated with nodes. N represents any base. "¬" indicates negation and "+" indicates one or more repeated times. The score functions $s(i, j, k)$ provide scores for sequences along the arcs. $l \in T(k)$ if there is a directed edge from k to l. $t(l, k)$ map to the possible outward arcs for node k. In this model, a single variable length codon state is used. In more sophisticated models there might be elaborate splice site states, states for initial, internal and final coding exons, promoter and polyadenylation sites, and reverse strand genes. The double circle is the start and end state.

The addition of length and state transition restrictions significantly improves running time, while ensuring that only meaningful parses are considered. In addition, software engineers for different systems have employed other tricks to improve the speed of gene prediction to approximately linear in the length of the input DNA sequence [9,46,55].

When the segment scores and transition scores are defined as log probabilities, which is easily derived from feature scoring methods such as weight matrices and Markov models, then we say that such a model of gene structure is a stochastic regular grammar or equivalently a HMM. (Note that sometimes the inclusion of variable length segments in the model is called a generalized HMM (GHMM) [44] or a state-duration HMM [60].) The dynamic program is then a maximum likelihood optimization:

$$\arg\max_{\Phi}(-\log P(X,\Phi)) \ .$$

Such models are called generative models because the score functions are decomposed into conditional probability terms of the form $-\log P(x_i \dots x_j|q_k)$, corresponding to the $s(i,j,k)$ score function, and $-\log P(q_l|q_k)$, the $t(l,k)$ transition score. It is sometimes convenient to describe HMMs as generating the sequence X via a random walk through the finite state machine. The decoding step is, then, the prediction of the most likely random path, Φ, that generated the observed data (see also Chapter 3).

Almost all of the successful, modern gene finders are based on this HMM framework including the most widely used *ab initio* gene finders GENSCAN [9] and FGENES [65]. Furthermore, we will shortly see that improvements to gene finders with respect to the inclusion of homologous protein, aligned cDNA and orthologous DNA are all extensions of this basic HMM framework.

An additional advantage of the probabilistic framework is that the score functions can be learned systematically using standard learning procedures, i.e. a maximum likelihood optimization using the forward–backward algorithm [60]. In practice, the parameters for the different score functions are trained independently in most gene-finding programs, but HMMGene [43] is a notable exception that achieves good performance. In addition, using the same algorithm for a test sequence it is possible to obtain the score of any single feature (such as an internal coding exon) in the context of all possible parses that might contain it. Studies have shown that these scores are meaningful metrics for ranking the confidence of different segments of a prediction [9,46,63].

7 Performance Comparisons

Performance of different gene finders has been assessed by several researchers including studies by Reese and coworkers [62] on *Drosophila* and Rogic and coworkers [63] on mammalian sequences. First, in Reese and coworkers, a 2.9-Mb contiguous DNA sequence was subjected to automated analysis by a battery of gene-finding programs and compared with the gene structures from a careful manual curation. Assessing false-positive rate (over-prediction) in this test was problematic because full-length gene structures were known with certainty for only a fraction of genes, full-length cDNAs were rejected if they did not meet certain automatic criteria such as having a good splice site score and for those uncertain genes there was a serious bias because automated gene finder predictions had been used by the manual curators to guide their annotations. Five *ab initio* gene finders were tested and standard evaluation statistics were collected (using the same metrics as in Ref. [12]). Table 2 presents the performance predicting individual exons and the entire CDS from start to stop codon. The only strong conclusion that can be made is that HMM-based gene finders (FGENES, Genie and HMMGene) perform comparably and superior to the other methods. We also know from these tests and others that gene finders naturally perform better when trained with examples from the organism being tested or related species [78].

Table 2 *Ab initio* performance for the *Adh* locus in *Drosophila*

	FGENES (v1/v2/v3)			GeneID (v1/v2)		Genie	HMMGene	Grail
Exon								
Sn	0.65	0.44	0.75	0.27	0.58	0.70	0.68	0.42
Sp	0.49	0.68	0.24	0.29	0.34	0.57	0.53	0.41
Missing	0.11	0.46	0.06	0.54	0.21	0.08	0.05	0.24
Wrong	0.32	0.17	0.53	0.48	0.47	0.17	0.20	0.29
CDS								
Sn	0.30	0.09	0.37	0.02	0.26	0.40	0.35	0.14
Sp	0.27	0.18	0.10	0.05	0.10	0.29	0.30	0.12
Missing	0.09	0.35	0.09	0.44	0.14	0.05	0.07	0.16
Wrong	0.24	0.25	0.52	0.22	0.30	0.11	0.15	0.24

Sn refers to the fraction of known genes that were predicted exactly correct. Sp is the fraction of predicted genes that were exactly correct. Missing is the fraction of known genes with no overlapping prediction. Wrong is the fraction of predictions that do not overlap annotated genes. High Sn and Sp and low Missing and Wrong values are better. Sn and Missing were determined from a different test set than Sp and Wrong. FGENES and GeneID were run under multiple parameter settings to produce different sensitivity/specificity trade-offs. See Ref. [62] for details. Importantly, different versions of these programs have typically been developed since these tests, so conclusions should be qualitative regarding methodology only.

Recognizing that biases could exist in gene prediction programs if testing data included gene structures used in training, Rogic and coworkers assessed the performance of seven *ab initio* gene finders on 195 mammalian gene structures that were submitted to GenBank after the programs were released. In the overall results shown in Table 3, the ranges of performance measures is comparable to that for invertebrates and the gene finder HMMGene shows a

Table 3 *Ab initio* performance for mammalian genes [63]

	FGENES	GeneMark	Genie	GENSCAN	HMMGene	Morgan	MZEF
Exon							
Sn	0.67	0.53	0.71	0.70	0.76	0.46	0.58
Sp	0.67	0.54	0.70	0.70	0.77	0.41	0.59
Missing	0.12	0.13	0.19	0.08	0.12	0.20	0.32
Wrong	0.09	0.11	0.11	0.09	0.07	0.28	0.23

Measures are interpreted as in Table 2.

significant advantage over other methods. However, to complicate matters, Rogic and coworkers also found that gene finder performance was frequently highly dependent on gene or genome characteristics such as type of CDS exon (initial, internal or terminal) and G + C content.

A third study, by Guigo and coworkers, has shown that gene finders perform notably worse on long DNA sequences than for the short test sequences that contain only one gene found in assessment studies [28], so mammalian performance shown here is probably an upper bound and should be evaluated only relatively.

8 Using Homology

A second class of gene finders is those that take advantage of homologous sequences from databases of cDNA, DNA and protein sequences. Alignments of cDNA indicate the exon–intron structure. Conserved sequences between orthologous chromosomes indicates functional DNA, i.e. regulatory and protein-coding sequences, and protein–DNA similarity identifies putative CDS.

8.1 cDNA Clustering and Alignments

The gold standard for gene structures are derived from the alignment of cDNAs (complementary DNA from mRNA) to DNA. A full-length cDNA requires only the identification of the CDS, which is typically assumed to be the largest ORF. However, full-length cDNAs are rare. Instead, tens of millions of cDNA fragments called expressed sequence tags (ESTs) with lengths of several hundred bases have been deposited in GenBank. These sequences are typically random sequencing reads from the $3'$ or $5'$ ends of libraries of cloned full-length or partial mRNAs. The primary difficulties with ESTs are the relatively short size, the frequent sequencing errors and the sheer number of such sequences.

Occasionally ESTs are analyzed individually in the hope of identifying fragments of coding regions. For this purpose, specialized HMMs similar to profile HMMs (see Chapter 3) have been developed to identify the reading frame with the highest coding potential while allowing for frame shifts that interrupt the CDS [35]. The methods employ similar, but simpler, Markov model scoring metrics and state machines than those used for gene finders in DNA (Figure 6).

Most EST analysis is based on assembling multiple EST sequences to form longer cDNA sequences. If no genome sequence is available, then rapid clustering methods are often employed to generate groups of homologous

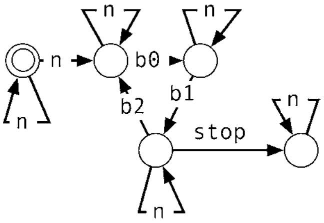


Figure 6 A finite state automaton for labeling the CDS (codons of "b0 b1 b2") in EST sequences allowing for frameshifts. A function such as the fifth-order Markov model is used for the scoring of the b0, b1 and b2 arcs.

ESTs [10]. These groups are then input to a conventional fragment assembler (see Chapter 2). This approach typically generates undesirable chimeric or partial assemblies and alternative isoforms can cause havoc.

In the conventional assembly approach, each EST contributes to just one assembled cDNA, but if multiple isoforms exist, then an EST should naturally be a part of multiple cDNA assemblies. A splice graph captures all of the possible isoforms implied by a set of ESTs [33]. In the splice graph, a virtual genome sequence is deduced, nodes are positions in the genome, and arcs connect positions that are adjacent in aligned ESTs. Figure 7 shows an example visualization of a splice graph. The splice graph makes clear that the number of possible isoforms can, in the worst case, be exponential to the number of exons.

With a completed genome sequence, cDNAs can be aligned to the DNA, which serves as a template, and the gene structure can be delineated by the connected set of overlapping, aligned ESTs. This is superior to the previous cluster-based transcript assembly because (i) errors in ESTs can be corrected by comparison with DNA, (ii) chimeras are less likely and (iii) the genome is directly annotated providing exon–intron structure.

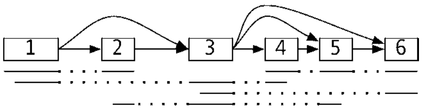


Figure 7 Splice graph. Nodes are positions along the horizontal axis, representing a DNA sequence. (When no genome sequence is available, the DNA sequence is virtual and ambiguous, i.e. implied exclusively by the differences between homologous ESTs.) Nodes are connected based on EST evidence. Those nodes adjacent on the DNA are merged into exons (numbered boxes). Arcs between boxes are introns. Assembled ESTs are shown below the splice graph. Dotted lines show where an EST spans across a DNA gap. Note that the genomic DNA is not necessary to build a splice graph. However, errors in the EST sequences can easily introduce false variants. To address this, multiple ESTs are usually required to confirm alternative splicings.

Highly accurate programs for cDNA–DNA alignment that include special handling for small exons, accurate splice site definition and EST errors are now available, most notably GMAP [79] or BLAT [38]. Imposing strict alignment criteria is usually advisable such as requiring that at least 90% of the EST be aligned with 95% identity in the aligned region. Even stricter alignment criteria are common, such as requiring that all intron gaps in an alignment conform to consensus dinucleotide splice sites and span a minimal number of genomic bases.

EST sequences are obtained from the $3'$ and $5'$ ends of cDNA clone inserts (and sometimes random internal positions). Due to the construction of the vector sequence containing the cDNA insert, sequencing of the two ends occurs on opposite strands of the insert. By convention these sequences are deposited in GenBank without reverse complementing and so usually a $5'$ EST sequence is of the sense strand and a $3'$ EST is of the anti-sense strand. Therefore, the orientation of a gene on the DNA can be inferred by comparing the EST read direction and the orientation of the sequence in the EST–DNA alignment. However, the labeling of the read direction and the strand that is submitted to the database is only a convention and there are frequent errors in the EST database, mostly among older database entries due to lane shifts from gel-based sequencing machines [1]. Other characteristics can be used to infer orientation of the EST including comparisons to other aligned sequences, presence of a poly-A tail (or poly-T prefix), presence of a polyadenylation signal and, most effectively, the consensus dinucleotides in the splice sites, if the EST splices. Shendure and Church [69] describe one laboratory's orientation procedure although no single software program currently exists to perform this orientation. (A lingering problem remains that EST–DNA alignments may indicate anti-sense transcription – a phenomenon that has been increasingly documented [81]; however, conventional gene modeling and genome annotation prohibit overlapping transcripts.)

Once individual ESTs are oriented and aligned, longer transcripts can be derived by merging gene structures implied by overlapping ESTs. As with EST analysis without a genome sequence, conflicting alignments can imply a large number of putative alternative splice forms. The PASA program is one of several programs that can be used to generate a a set of such putative transcripts from EST–DNA alignments [31,37]. In PASA, a dynamic program, is employed to assemble a minimal set of unique transcripts from compatible EST alignments, i.e. those ESTs that agree in all of their inferred splicings.

Not all genes will be present in EST libraries because differentially expressed genes may be absent or expressed at low levels in the sampled tissues. Furthermore, due to limitations in full-length cDNA cloning and the long lengths of some transcripts, many genes are not fully covered by EST sequences. Thus, assembly of ESTs tends to result in fragmented, partial transcripts.

One method that has been used to specifically deal with the incomplete EST information is Genie [46]. The method assumes that assembled transcripts from EST–DNA alignments define true, but incomplete, gene structures and so the *ab initio* gene-finding algorithm described in Section 6.3 is employed only in the breaks between alignments. This can be achieved in a straightforward manner by modifying the $t(\dot{;})$ transcription score function to prohibit transitioning into some states at specific positions within the sequence depending on the EST–DNA alignment evidence (e.g. transitioning into an exon state in the middle of an EST-defined intron is prohibited in the dynamic program).

Lastly, it is possible to leverage the EST data to infer gene bounds even when only incomplete EST alignments exist using EST mate pairs. For some cDNA inserts both the $5'$ and $3'$ ends have been sequenced. When both ESTs are aligned to the same chromosome, within a reasonable genomic distance, and compatibly ordered and oriented, then one can infer that the entire region between the mate pairs corresponds to a single primary transcript. In a similar way as above, an *ab initio* gene finder can be constrained to predict exactly one primary transcript in the defined region.

8.2 Orthologous DNA

When two organisms are sufficiently similar to identify and align orthologous genomic sequence, but sufficiently distant so that nonfunctional DNA has mutated, then the comparative analysis of the two genomes can be directly applied to gene finding (see also Chapter 37). Organisms such as chicken and mouse are of a reasonable evolutionary distance from human to support this sort of comparison. The key assumptions are that the number and approximate content of CDS regions between the two species are well conserved, while introns, UTRs and intergenic DNA have drifted significantly from the common ancestor. Thus, if the conserved regions can be identified, then they are most likely coding regions and so the gene-finding problem is to combine these conserved CDS segments into a more complete gene structure.

There are two main approaches to the problem: *ad hoc* weighting schemes and principled pair HMMs. Twinscan [39] and SGP2 [56] are examples of the former method. For example, in Twinscan, the dynamic program corresponding to the gene grammar described in Section 6.3 is augmented with scores from BLAST sequence similarity matches between the two genomes. In other words, for any candidate region $x_i \dots x_j$ in a CDS state, q_k, the score function $s(i, j, k)$ is improved according to the quality of the genome–genome

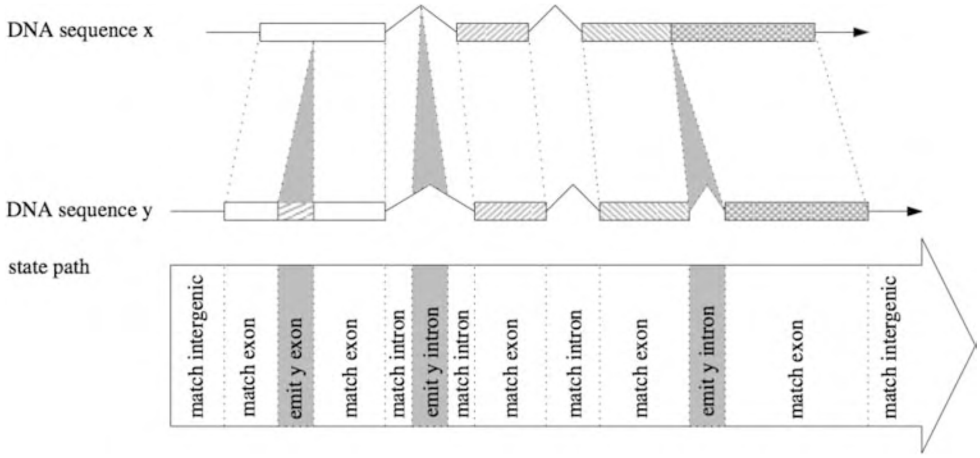


Figure 8 Pair HMM. Two DNA sequences are generated simultaneously from left to right. At each step, a subsequence (possibly of length zero) is emitted for each genome. The score for the pairs of subsequences is based on the local statistics like a conventional *ab initio* gene finder as well as the similarity of the two subsequences. (From Meyer and Durbin [51].)

alignment in that region. Thus, the method performs gene finding on one genome sequence using the second genome as evidence.

The second, more elegant, approach is to simultaneously align and label both genome sequences according to a probabilistic model. DoubleScan [51] and SLAM [3] embody this class of gene finders. The approach, called a pair HMM, is a generalization of the *ab initio* method and is best understood by considering an HMM-based gene finder as a generative model that produces labeled sequences of DNA, as previously described in Section 6.3. In a conventional HMM, one or more nucleotides are emitted for each state; in a pair HMM two sequences of nucleotides are emitted for each state. To address asymmetries such as inserts in one genome additional states must be added to allow for null string emissions in one genome. Figure 8 shows a diagram of the generative process.

Our score function $s(i, j, k)$ is extended to consider the scores of pairs of segments, i.e. $s(i, j, m, n, k)$ is the score for simultaneously emitting DNA sequences $x_i \ldots x_j$ and $y_m \ldots y_n$ in state q_k. The dynamic program for a pair HMM is not much different from the conventional single sequence HMM although the pair HMM must consider, theoretically, all possible segments $x_i \ldots x_j$ and $y_m \ldots y_n$ in every possible state, which adds a very significant computational burden. In practice this computational burden may not be worth the investment since the *ad hoc* score enhancement methods are fast and have been shown to perform well.

Conserved noncoding sequences (CNS), which are usually regulatory sequences, are often a problem for comparative modeling methods because they are interpreted as coding. In some implementations special CNS states are introduced and, in theory, if the CNS sequences lack sufficient coding potential, then they will be labeled CNS instead of CDS. In practice, mispredicting CNS remains a challenge, particularly for highly conserved genes.

The comparative method has also been extended to model the proper phylogenetic distance among three or more genomes in a phylo-HMM [70] and also to successfully annotate very closely related species such as among primates [50].

8.3 Protein Homology

The simplest application of protein homology is the identification of putative CDS subsequences from sequence alignment. For example, BLASTX [24] performs six-frame translation of a DNA query sequence and rapidly identifies those regions that are similar to known proteins. The user must assemble the fragmentary evidence.

The most elegant and specific use of protein homology is employed by the GeneWise [6] program, which merges the profile HMM used in protein remote homology searching (see Chapter 11) and the gene finding HMM model described in Section 6.3 into a unified DNA–protein alignment. The method is a pair HMM, similar to those used in comparative genomic analysis, but in this case the model is more complex because the two generated sequences use different alphabets, and additional constraints must be included to ensure proper pairing of amino acids and codons according to the genetic code. The model also includes a basic set of splice site recognition states to allow for the alignment of the protein sequence across introns (similar to cDNA–DNA alignment) as well as nucleotide insertion and deletion states to allow for errors and frame shifts.

GeneWise produces only partial gene structures corresponding to the region of protein alignment on the DNA. However, importantly, such alignments provide highly accurate predictions when a sufficiently close homolog is available. Moreover, the prediction of splice sites is particularly good due to the constraint of the alignment of the protein across introns.

HMM gene-finding programs such as FGENES++ [65] and Genie [45] employ a more *ad hoc* approach in which scores for coding features are artificially inflated when database similarities are found, in a similar manner as the comparative genome program Twinscan [39]. As a protein–DNA alignment improves, the score for labeling the DNA region as coding improves. In this way, a complete gene structure is predicted with protein homology evidence contributing, but it is not used exclusively nor is it required. Such an approach requires careful tuning of the contribution of protein homology to avoid overprediction or over extension of coding exons.

8.4 Integrative Methods

There has been much work on integrative methods of combining multiple gene finders and homology evidence, both principled and *ad hoc*, for whole genome analysis, but we do not review them in detail here. Several programs have attempted to integrate the predictions of multiple gene finders within a probabilistic model (e.g. Refs. [4, 53, 57]). Other programs provide an abstract framework of an HMM gene finder that allows a software developer to incorporate arbitrary feature scoring methods [34]. Most genome centers and informatics sites maintain "pipelines" for automated annotation. Many of these are not portable or are tightly bound to other institutional software infrastructure. Two noteworthy examples are the annotation pipelines of the NCBI (http://www.ncbi.nlm.nih.gov/genome/guide/build.html) and Ensembl [17]. Both include sophisticated and comprehensive methods for reliable whole genome gene prediction.

9 Pitfalls: Pseudogenes, Splice Variants and the Cruel Biological Reality

As this chapter concludes its tour of gene-finding methods, it is worth a brief mention of some of the unfortunate difficulties that make gene prediction a hard problem that is unlikely to be satisfactorily solved in the near term. The challenges almost all lie in the complexity of genome organization that is not (and often cannot be) modeled by the various gene-finding techniques [52]. Here are several issues:

- Exons can be extremely small – only a few nucleotides, which is insufficient to detect coding potential. Worse, re-splicing has been observed in which an exon is entirely removed.
- Noncanonical splice sites are expected to occur, on average, about once every 10 genes in the genome; however, with few exceptions, methods assume GT/AG splice sites.
- Pseudogenes are numerous in many organisms including the human. There are specialized programs to detect retrotranscribed and nonfunctional genes (e.g. Ref. [15]), but young pseudogenes are often quite difficult to distinguish by sequence statistics.

- Alternative splicing is prolific in higher eukaryotes. It is estimated that 60% of human genes have multiple isoforms. We saw in Section 8.1 that cDNA methods can be used to enumerate possible splice variants. HMM-based methods like those we learned here can be used to generate suboptimal parses that sometimes represent alternative isoforms [14]. Recent predictive methods have been developed for detecting alternative splice sites [19]. Nevertheless, alternative splicing remains a serious impediment to automated genome annotation.
- Untranslated exons make up a large fraction of the typical gene, yet there is very little, if any, signal differentiating UTRs from intergenic DNA.
- The size of most genomes implies that gene-like patterns are likely to occur frequently in intergenic DNA. In order to control false-positive rates, gene-finding programs must also sacrifice true positives. Studies of gene finders in large DNA sequences reveal frequent overprediction [28].
- Anti-sense transcription [81], high rates of transcription outside of known protein-coding genes [36] and large classes of small noncoding RNAs have been observed [54]. Besides causing difficulties in the use of cDNA–DNA alignment evidence for gene finding, these findings emphasize that important, functional, nonprotein-coding genes are being transcribed at high rates and that new classes of genes may yet be discovered. Alternatively, the high rates of transcription observed by Kampa and coworkers [36] also suggests that the genome is less of a programmed machine and more of a stochastic process in which nonfunctional transcription may be occurring at high levels. Such "noise" in the system is insurmountable using sequence analysis alone.

As a result of these and other complications, manual genome annotation is expected to remain the definitive source for gene structures for a long time. An excellent source of manual curations is the VEGA system [49].

10 Further Reading

Chapter 3 of this book introduces HMMs. Further background on HMMs and probabilistic modeling of gene sequences – the techniques that dominate gene finding – is best found in the book *Biological Sequence Analysis* [20] and Rabiner's oft-cited tutorial [60]. An excellent gene-finding bibliography is maintained by Wentian Li (http://www.nslij-genetics.org/gene/). The primary literature for generalized HMMs (e.g. GENSCAN [9] and Genie [44,46]) and comparative gene finders (e.g. DoubleScan [51], SLAM [3], and Shadower

[50]) is particularly good for new readers. There are many reviews of gene-finding techniques and Zhang's [84] is a relatively recent good one. The website www.genefinding.org is also a useful resource for developers.

References

1 Aaronson, J., B. Eckman, R. Blevins, J. Borkowski, J. Myerson, S. Imran and K. Elliston. 1996. Toward the development of a gene index to the human genome: an assessment of the nature of high-throughput EST sequence data. Genome Res. **6**: 829–45.

2 Agarwal, P. and V. Bafna. 1998. The ribosome scanning model for translation initiation: implications for gene prediction and full-length cDNA detection. Proc. ISMB **6**: 2–7.

3 Alexandersson, M., S. Cawley and L. Pachter. 2003. SLAM: cross-species gene finding and alignment with a generalized pair hidden Markov model. Genome Res. **13**: 496–502.

4 Allen, J., M. Pertea and S. Salzberg. 2004. Computational gene prediction using multiple sources of evidence. Genome Res. **14**: 142–8.

5 Berget, S. 1995. Exon recognition in vertebrate splicing. J. Biol. Chem. **270**: 2411–4.

6 Birney, E., M. Clamp and R. Durbin. 2004. GeneWise and Genomewise. Genome Res. **14**: 988–95.

7 Borodovsky, M. and J. McIninch. 1993. GeneMark: parallel gene recognition for both DNA strands. Comput. Chem. **17**: 123–33.

8 Brunak, S., J. Engelbrecht and S. Knudsen. 1991. Prediction of human mRNA donor and acceptor sites from the DNA sequence. J. Mol. Biol. **220**: 49–65.

9 Burge, C. and S. Karlin. 1997. Prediction of complete gene structures in human genomic DNA. J. Mol. Biol. **268**: 78–94.

10 Burke, J., D. Davison and W. Hide. 1999. d2_cluster: a validated method for clustering EST and full-length cDNA sequences. Genome Res. **9**: 1135–42.

11 Burset, M., I. Seledtsov and V. Solovyev. 2000. Analysis of canonical and non–canonical splice sites in mammalian genomes. Nucleic Acids Res. **28**: 4364–75.

12 Burset, M. and R. Guigo. 1996. Evaluation of gene structure prediction programs. Genomics **34**: 353–67.

13 Cai, D., A. Delcher, B. Kao and S. Kasif. 2000. Modeling splice sites with Bayes networks. Bioinformatics **16**: 152–8.

14 Cawley, S. and L. Pachter. 2003. HMM sampling and applications to gene finding and alternative splicing. Bioinformatics **19** (Suppl. 2): II36–41.

15 Coin, L. and R. Durbin. 2004. Improved techniques for the identification of pseudogenes. Bioinformatics **20** (Suppl. 1): I94–I100.

16 Consortium, T. 1998. Genome sequence of the nematode *C. elegans*: a platform for investigating biology. Science **282**: 2012–8.

17 Curwen, V., E. Eyras, T. Andrews, L. Clarke, E. Mongin, S. Searle and M. Clamp. 2004. The Ensembl automatic gene annotation system. Genome Res. **14**: 942–50.

18 Dong, S. and D. Searls. 1994. Gene structure prediction by linguistic methods. Genomics **23**: 540–51.

19 Dror, G., R. Sorek and R. Shamir. 2005. Accurate identification of alternatively spliced exons using support vector machine. Bioinformatics **21**: 897–901.

20 Durbin, R., S. Eddy, A. Krogh and G. Mitchison. 1998. *Biological Sequence Analysis.* Cambridge University Press, Cambridge.

21 Fickett, J. and A. Hatzigeorgiou. 1997. Eukaryotic promoter recognition. Genome Res. **7**: 861–78.

22 Fickett, J. and C. Tung. 1992. Assessment of protein coding measures. Nucleic Acids Res **20**: 6441–50.

23 Forney, G. 1973. The Viterbi algorithm. Proc. IEEE **61**: 268–78.

24 Gish, W. and D. States. 1993. Identification of protein coding regions by database similarity search. Nat. Genet. **3**: 266–72.

25 Graber, J., G. McAllister and T. Smith. 2002. Probabilistic prediction of *Saccharomyces cerevisiae* mRNA 3′-processing sites. Nucleic Acids Res. **30**: 1851–8.

26 Gribskov, M., J. Devereux and R. Burgess. 1984. The codon preference plot: graphic analysis of protein coding sequences and prediction of gene expression. Nucleic Acids Res. **12**: 539–49.

27 Grosjean, H. and W. Fiers. 1982. Preferential codon usage in prokaryotic genes: the optimal codon-anticodon interaction energy and the selective codon usage in efficiently expressed genes. Gene **18**: 199–209.

28 Guigo, R., P. Agarwal, J. Abril, M. Burset and J. Fickett. 2000. An assessment of gene prediction accuracy in large DNA sequences. Genome Res. **10**: 1631–42.

29 Guigo, R. and J. Fickett. 1995. Distinctive sequence features in protein coding genic non-coding, and intergenic human DNA. J. Mol. Biol. **253**: 51–60.

30 Guo, M. and S. Mount. 1995. Localization of sequences required for size-specific splicing of a small *Drosophila* intron *in vitro*. J. Mol. Biol. **253**: 426–37.

31 Haas, B., A. Delcher, S. Mount, J. Wortman et al. 2003. Improving the Arabidopsis genome annotation using maximal transcript alignment assemblies. Nucleic Acids Res. **31**: 5654–66.

32 Hawkins, J. 1988. A survey on intron and exon lengths. Nucleic Acids Res. **16**: 9893–908.

33 Heber, S., M. Alekseyev, S. Sze, H. Tang and P. Pevzner. 2002. Splicing graphs and EST assembly problem. Bioinformatics **18** (Suppl. 1): S181–8.

34 Howe, K., T. Chothia and R. Durbin. 2002. GAZE: a generic framework for the integration of gene-prediction data by dynamic programming. Genome Res. **12**: 1418–27.

35 Iseli, C., C. Jongeneel and P. Bucher. 1999. ESTScan: a program for detecting, evaluating, and reconstructing potential coding regions in EST sequences. Proc. ISMB **7**: 138–48.

36 Kampa, D., J. Cheng, P. Kapranov, et al. 2004. Novel RNAs identified from an in-depth analysis of the transcriptome of human chromosomes 21 and 22. Genome Res. **14**: 331–42.

37 Kan, Z., E. Rouchka, W. Gish and D. States. 2001. Gene structure prediction and alternative splicing analysis using genomically aligned ESTs. Genome Res. **11**: 889–900.

38 Kent, W. 2002. BLAT – the BLAST-like alignment tool. Genome Res. **12**: 656–64.

39 Korf, I., P. Flicek, D. Duan and M. Brent. 2001. Integrating genomic homology into gene structure prediction. Bioinformatics **17** (Suppl. 1): S140–8.

40 Kozak, M. 1987. An analysis of 5′-noncoding sequences from 699 vertebrate messenger RNAs. Nucleic Acids Res. **15**: 8125–48.

41 Kozak, M. 1991. Structural features in eukaryotic mRNAs that modulate the initiation of translation. J. Biol. Chem. **266**: 19867–70.

42 Krogh, A., M. Brown, I. Mian, K. Sjolander and D. Haussler. 1994. Hidden Markov models in computational biology. Applications to protein modeling. J. Mol. Biol. **235**: 1501–31.

43 Krogh, A. 1997. Two methods for improving performance of an HMM and their application for gene finding. Proc. ISMB **5**: 179–86.

44 Kulp, D., D. Haussler, M. Reese and F. Eeckman. 1996. A generalized hidden Markov model for the recognition of human genes in DNA. Proc. ISMB **4**: 134–42.

45 Kulp, D., D. Haussler, M. Reese and F. Eeckman. 1997. Integrating database homology in a probabilistic gene structure model. Pac. Symp. Biocomput. **2**: 232–44.

46 KULP, D. 2003. Protein-coding gene structure prediction using generalized hidden Markov models. University of California. PhD Dissertation.

47 LEGENDRE, M. AND D. GAUTHERET. 2003. Sequence determinants in human polyadenylation site selection. BMC Genomics **4**: 7.

48 LI, H. AND T. JIANG. 2004. A class of edit kernels for SVMs to predict translation initiation sites in eukaryotic mRNAs. Proc. RECOMB **8**: 262–71.

49 LOVELAND, J. 2005. VEGA, the genome browser with a difference. Brief. Bioinform. **6**: 189–93.

50 MCAULIFFE, J., L. PACHTER AND M. JORDAN. 2004. Multiple-sequence functional annotation and the generalized hidden Markov phylogeny. Bioinformatics.

51 MEYER, I. AND R. DURBIN. 2002. Comparative *ab initio* prediction of gene structures using pair HMMs. Bioinformatics **18**: 1309–18.

52 MOUNT, S. 2000. Genomic sequence, splicing, and gene annotation. Am. J. Hum. Genet. **67**: 788–92.

53 MURAKAMI, K. AND T. TAKAGI. 1998. Gene recognition by combination of several gene-finding programs. Bioinformatics **14**: 665–75.

54 OTA, T., Y. SUZUKI, T. NISHIKAWA, ET AL. 2004. Complete sequencing and characterization of 21,243 full–length human cDNAs. Nat. Genet. **36**: 40–5.

55 PARRA, G., E. BLANCO AND R. GUIGO. 2000. GeneID in *Drosophila.* Genome Res. **10**: 511–5.

56 PARRA, G., P. AGARWAL, J. ABRIL, T. WIEHE, J. FICKETT AND R. GUIGO. 2003. Comparative gene prediction in human and mouse. Genome Res. **13**: 108–17.

57 PAVLOVIC, V., A. GARG AND S. KASIF. 2002. A Bayesian framework for combining gene predictions. Bioinformatics **18**: 19–27.

58 PEDERSEN, A. AND H. NIELSEN. 1997. Neural network prediction of translation initiation sites in eukaryotes: perspectives for EST and genome analysis. Proc. ISMB **5**: 226–33.

59 PERTEA, M., X. LIN AND S. SALZBERG. 2001. GeneSplicer: a new computational method for splice site prediction. Nucleic Acids Res. **29**: 1185–90.

60 RABINER, L. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**: 257–86.

61 REESE, M., F. EECKMAN, D. KULP AND D. HAUSSLER. 1997. Improved splice site detection in Genie. J. Comput. Biol. **4**: 311–23.

62 REESE, M., G. HARTZELL, N. HARRIS, U. OHLER, J. ABRIL AND S. LEWIS. 2000. Genome annotation assessment in *Drosophila melanogaster*. Genome Res. **10**: 483–501.

63 ROGIC, S., B. OUELLETTE AND A. MACKWORTH. 2002. Improving gene recognition accuracy by combining predictions from two gene–finding programs. Bioinformatics **18**: 1034–45.

64 SALAMOV, A., T. NISHIKAWA AND M. SWINDELLS. 1998. Assessing protein coding region integrity in cDNA sequencing projects. Bioinformatics **14**: 384–90.

65 SALAMOV, A. AND V. SOLOVYEV. 2000. *Ab initio* gene finding in *Drosophila* genomic DNA. Genome Res. **10**: 516–22.

66 SALZBERG, S., M. PERTEA, A. DELCHER, M. GARDNER AND H. TETTELIN. 1999. Interpolated Markov models for eukaryotic gene finding. Genomics **59**: 24–31.

67 SALZBERG, S. 1997. A method for identifying splice sites and translational start sites in eukaryotic mRNA. Comput. Appl. Biosci. **13**: 365–76.

68 SEARLS, D. 1992. The Linguistics of DNA. Am. Sci. **80**: 579–91.

69 SHENDURE, J. AND G. CHURCH. 2002. Computational discovery of sense–antisense transcription in the human and mouse genomes. Genome Biol. **3**: 1–14.

70 SIEPEL, A. AND D. HAUSSLER. 2004. Combining phylogenetic and hidden Markov models in biosequence analysis. J. Comput. Biol. **11**: 413–28.

71 Snyder, E. and G. Stormo. 1993. Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks. Nucleic Acids Res. **21**: 607–13.

72 Solovyev, V., A. Salamov and C. Lawrence. 1994. Predicting internal exons by oligonucleotide composition and discriminant analysis of spliceable open reading frames. Nucleic Acids Res. **22**: 5156–63.

73 Sonnenburg, S., G. Ratsch, A. Jagota and M. Klaus-Robert. 2002. New methods for splice site recognition. In Proc. Int. Conf. on Artificial Neural Networks, Madrid: 329–36.

74 Staden, R. 1984. Computer methods to locate signals in nucleic acid sequences. Nucleic Acids Res. **12**: 505–19.

75 Staden, R. 1990. Finding protein coding regions in genomic sequences. Methods Enzymol. **183**: 163–80.

76 Stormo, G. 1990. Consensus patterns in DNA. Methods Enzymol. **183**: 211–21.

77 Tabaska, J. and M. Zhang. 1999. Detection of polyadenylation signals in human DNA sequences. Gene **231**: 77–86.

78 Waterston, R., K. Lindblad-Toh, E. Birney, et al. 2002. Initial sequencing and comparative analysis of the mouse genome. Nature **420**: 520–62.

79 Wu, T. and C. Watanabe. 2005. GMAP: a genomic mapping and alignment program for mRNA and EST sequences. Bioinformatics **21**: 1859–75.

80 Yan, J. and T. Marr. 2005. Computational analysis of 3′-ends of ESTs shows four classes of alternative polyadenylation in human, mouse, and rat. Genome Res. **15**: 369–75.

81 Yelin, R., D. Dahary, R. Sorek, et al. 2003. Widespread occurrence of antisense transcription in the human genome. Nat. Biotechnol. **21**: 379–86.

82 Zhang, L. and L. Luo. 2003. Splice site prediction with quadratic discriminant analysis using diversity measure. Nucleic Acids Res. **31**: 6214–20.

83 Zhang, M. 1997. Identification of protein coding regions in the human genome by quadratic discriminant analysis. Proc. Natl. Acad. Sci. USA **94**: 565–8.

84 Zhang, M. 2002. Computational prediction of eukaryotic protein–coding genes. Nat. Rev. Genet. **3**: 698–709.

85 Zien, A., G. Ratsch, S. Mika, B. Scholkopf, T. Lengauer and K. Muller. 2000. Engineering support vector machine kernels that recognize translation initiation sites. Bioinformatics **16**: 799–807.

6
Analyzing Regulatory Regions in Genomes

Thomas Werner

1 General Features of Regulatory Regions in Eukaryotic Genomes

Regulatory regions share several common features despite their obvious divergence in sequence. Most of these common features are not evident directly from the nucleotide sequence, but result from the restraints imposed by functional requirements. Therefore, understanding of the major components and events during the formation of regulatory DNA–protein complexes is crucial for the design and evaluation of algorithms for the analysis of regulatory regions. Transcription initiation from polymerase II (Pol II) is the best understood example so far and will be a major focus of this chapter. However, the mechanisms and principles revealed from promoters are mostly valid for other regulatory regions as well.

Algorithms for the analysis and recognition of regulatory regions draw from the underlying biological principles, to some extent, in order to generate suitable computational models. Therefore, a brief overview over the biological requirements and mechanisms is necessary to understand what are the strengths and weaknesses of the individual algorithms. The choice of parameters and implementation of the algorithms largely control the sensitivity and speed of a program. The specificity of software recognizing regulatory regions in DNA is determined, to a large extent, by how closely the algorithm follows what will be called the biological model from hereon. Several overviews of this topic have been published [27, 96].

1.1 General Functions of Regulatory Regions

The biological functionality of regulatory regions is generally not a property evenly spread over the regulatory region in total. Functional units usually are defined by a combination of defined stretches that can be delimited and possess an intrinsic functional property (e.g. binding of a protein or a curved DNA structure). Several functionally similar types of these stretches of DNA are already known and will be referred to as *elements*. Those elements are

Bioinformatics - From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

neither restricted to regulatory regions nor individually sufficient for the regulatory function of a promoter or enhancer. The function of the complete regulatory region is composed from the functions of the individual elements either in an additive manner (independent elements) or by synergistic effects (modules).

1.2 Most Important Elements in Regulatory Regions

Transcriptional regulation depends on sequence elements that are directly accessible from the genomic DNA sequence such as transcription factor (TF)-binding sites (TFBSs), repeats and hairpins (repeats that can form hairpin-like structures by self-complementarity). In addition, various elements not easily detectable in the sequence are important. Most of these affect chromatin structure and accessibility such as histone acetylation and methylation status as well as DNA methylation status. Such phenomena not directly linked to the local DNA sequence are usually summarized under epigenetic effects.

1.3 TFBSs

Binding sites for specific proteins are most important among the sequence elements. They consist of about 10–30 nucleotides, not all of which are equally important for protein binding. As a consequence, individual protein-binding sites vary in sequence, even if they bind to the same protein. There are nucleotides contacted by the protein in a sequence-specific manner, which are usually the best-conserved parts of a binding site. Different nucleotides are involved in DNA backbone contacts, i.e. contacting the sugar-phosphate framework of the DNA helix (not sequence specific as they do not involve the bases A, G, C or T). There are also internal “spacers” not contacted by the protein at all. In general, protein-binding sites exhibit enough sequence conservation to allow for the detection of candidates by a variety of sequence similarity-based approaches. Potential binding sites can be found almost all over the genome and are not restricted to regulatory regions. Quite a number of binding sites outside regulatory regions are also known to bind their respective binding proteins [57]. Therefore, the abundance of predicted binding sites is not just a shortcoming of the detection algorithms, but reflects biological reality. Often it is not possible either to identify individual binding proteins as they might bind as part of multi-protein complexes [68]. This illustrates another important point: TF binding *in vivo* is usually context dependent. The isolated TF will bind to a cognate site quite differently if brought together in a reaction tube as naked protein and oligonucleotide probe than *in vivo* where adaptive DNA structure and a host of other proteins are present. As became evident from several chromatin immunoprecipitation

(ChIP) studies, even *in vivo* binding of a TF does not automatically imply a function in transcription control as was found in a genome-wide study which identified many more cAMP response element-binding protein (CREB)-binding sites than CREB-regulated genes [45].

1.4 Sequence Features

Regulatory DNA also contains several features not directly resulting in recognizable sequence conservation. For example, two copies of a direct repeat (approximate or exact) are conserved in sequence with respect to each other, but different direct repeats are not similar in sequence at all. Nevertheless, direct repeats are quite common within regulatory DNA regions. They consist either of short sequences, which are repeated twice or more frequently within a short region, or they can be complex repeats, which repeat a pattern of two or more elements. (More details on sequence repeats and how to detect them can be found in Chapter 7.) Repeat structures are often associated with enhancers. Enhancers are DNA structures that enhance transcription over a distance without being promoters themselves. One example of a highly structured enhancer is the interleukin-2 enhancer [74]. Other sequence features that are hard to detect by computer methods include the relatively weak nucleosomal positioning signals [46], DNA stretches with intrinsic three-dimensional (3-D) structures (like curved DNA, e.g. Ref. Ref. [81]), methylation signals (if there are definite signals for methylation at all) and other structural elements.

1.5 Structural Elements

Currently, secondary structures are the most useful structural elements with respect to computer analysis. Secondary structures are mostly known for RNAs (see Chapter 14) and proteins (see Chapter 9), but they also play important roles in DNA. DNA can form double hairpins called cruciform DNA representing the hairpin structures of RNA and can be important for transcriptional regulation [59]. Potential secondary structures can be easily determined and even scored via the negative enthalpy that should be associated with the actual formation of the hairpin (single-strand) or cruciform (double-strand) structure. Secondary structures are also not necessarily conserved in the primary nucleotide sequence, but are subject to strong positional correlation within the 3-D structure, i.e. the orientation of the double helix in space. Without any doubt 3-D aspects of DNA sequences are very important for the functionality of such regions. However, existing attempts to calculate such structures in reasonable time have met with mixed success and cannot be used for a routine sequence analysis at present. Part of that difficulty is that DNA structure can be quite flexible and structural changes are readily induced by interacting proteins [68].

1.6 Organizational Principles of Regulatory Regions

This section will mainly concentrate on eukaryotic polymerase II promoters, as they are currently the best-studied regulatory regions.

1.6.1 Overall Structure of Pol II Promoters

Promoters are DNA regions capable of specific initiation of transcription (start of RNA synthesis) and consist of three basic regions (see Section 1.6.3). The part determining the exact nucleotide for transcriptional initiation is called the core promoter, and is the stretch of DNA sequence where the RNA polymerase and its cofactors assemble on the promoter.

The region immediately upstream of the core promoter is called the proximal promoter and usually contains a number of TFBSs responsible for the assembly of an activation complex. This complex in turn recruits the polymerase complex. It is generally accepted that most proximal promoter elements are located within a stretch of about 250–500 nucleotides upstream of the actual transcription start site (TSS).

The third part of the promoter is located even further upstream and is called the distal promoter. This region usually regulates the activity of the core and the proximal promoter, and also contains TFBSs. However, distal promoter regions and enhancers exhibit no principal differences. If a distal promoter region acts position and orientation independent it is called an enhancer.

1.6.2 TFBS in Promoters

The TFBSs within promoters (and likewise most other regulatory sequences) do not show any general patterns with respect to location and orientation within the promoter sequences, although particular functionality may be associated with a specific location or association within the promoter [89].

Even functionally important binding sites for a specific TF may occur almost anywhere within a promoter. For example, functional activating protein 1 (AP-1, a complex of two TFs: one from the Fos and one from the Jun family)-binding sites can be located far upstream, as in the rat bone sialoprotein gene where an AP-1 site located about 900 nucleotides upstream of the TSS inhibits expression [97]. An AP-1 site located close to the TSS is important for the expression of Moloney murine leukemia virus [75]. Moreover, functional AP-1 sites have also been found inside exon 1 (downstream of the TSS) of the proopiomelanocortin gene [11] as well as within the first intron of the *fra-1* gene [6], both locations outside the promoter. Similar examples can be found for several other TF sites, illustrating why no general correlation of TF sites within specific promoter regions can be defined. TFBSs can be found virtually everywhere in promoters, but in individual promoters possible locations are much more restricted. A closer look reveals that the function of an AP-1-binding site often depends on the relative location and, especially, on the sequence context of the binding site. The AP-1 site in the above-mentioned rat bone sialoprotein gene overlaps with a set of glucocorticoid-responsive element (GRE, the DNA sequence that is bound by the glucocorticoid receptor which is a TF) half-sites (nuclear factor-binding sites are often composed of two almost identical half-sites separated by a spacer of a few nucleotides), which are crucial for the suppressive function.

The context of a TF site is one of the major determinants of its role in transcription control. As a consequence of context requirements, often TF sites are grouped together and such functional groups have been described in many cases. A systematic attempt at collecting synergistic or antagonistic pairs of TFBSs has been made with the COMPEL database [51]. In many cases, a specific promoter function (e.g. a tissue-specific silencer) will require more than two sites. Promoter subunits consisting of groups of TFBSs that carry a specific function independent of the promoter will be referred to as *promoter modules*. Arnone and Davidson originally gave a more detailed definition of promoter modules [1]. In summary, promoter modules contain several TFBSs which act together to convey a common function like tissue-specific expression. The organization of binding sites (and probably also of other elements) of a promoter module appears to be much more restricted than the apparent variety of TF sites and their distribution in the whole promoter suggests. Within a promoter module both sequential order and distance can be crucial for function, indicating that these modules may be the critical determinants of a promoter rather than individual binding sites. Promoter modules are always constituted by more than one binding site. Since promoters can contain several modules that may use overlapping sets of binding sites, the conserved context of a particular binding site cannot be determined from the primary sequence. The corresponding modules must be detectable separately before the functional modular structure of a promoter or any other regulatory DNA region can be revealed by computer analysis. One well-known general promoter module is the core promoter, which will be discussed in more detail below. However, the basic principles of modular organization are also true for most, if not all, other regulatory regions and are neither peculiar nor restricted to promoters.

1.6.3 Module Properties of the Core Promoter

The core promoter module can be defined functionally by its capability to assemble the transcription initiation complex and orient it specifically towards the TSS of the promoter [100], defining the exact location of the TSS. Various combinations of about four distinguishable core promoter elements that constitute a general core promoter can achieve this. This module includes the TATA box, the initiator region (INR), an upstream activating element and a downstream element (Figure 1). The TATA box is a basic transcription element, which is located about 20–30 nucleotides upstream of the actual TSS and is known to bind to the TATA box-binding protein (TBP). However, this is also where the straightforward definition of a core promoter module ends because not all four elements are required or some elements can be too variable to be recognizable by current computer tools.

The first group is made up of TATA box-containing promoters without a known initiator. Successful positioning of the initiation complex can start at the TATA box-containing promoters by the TFIID complex, which contains the TBP as well as several other factors. Together with another complex of general TFs, termed TFIIB, this leads to the assembly of an initiation complex [22]. If an appropriate upstream TFBS cooperates with the TATA box, no special initiator or downstream sequences might be required, which allows for the assembly of a functional core promoter module from just two of the four elements. This represents one type of a distinct core promoter that contains a TATA box, common among cellular genes in general.

The second group is TATA-less promoters with a functional initiator. As is known from a host of TATA-less promoters, however, the TATA box is by no means an essential element of a functional core promoter. An INR combined with a single upstream element has also been shown to be capable of specifically initiating transcription [41], although initiators cannot be clearly defined at the sequence level so far. Generally, a region of 10–20 nucleotides around the TSS is thought to represent the initiator. A remarkable array of four different upstream TF sites (SP1, AP-1, ATF or TEF1) was shown to confer inducibility by T-antigen to this very simple promoter, i.e. mediated transcriptional activation upon binding of T-antigen. T-antigen is a potent activating protein from a (simian) virus called SV40. This is an example of a TATA-less distinct promoter that can be found in several genes from the hematopoietic lineage (generating blood cells).

The third group is made up of a composite promoter consisting of both a TATA box and an initiator. This combination can be found in several viral promoters and it has been shown that an additional upstream TFBS can influence whether the TATA box or the initiator element will determine the promoter properties [21]. The authors showed that upstream elements can significantly increase the efficiency of the INR in this combination; in particular, SP-1 sites made the TATA box almost obsolete in their example. The combination of TATA box with an INR had the general effect of inducing resistance against the detrimental effects of a TFIIB mutant, which interfered

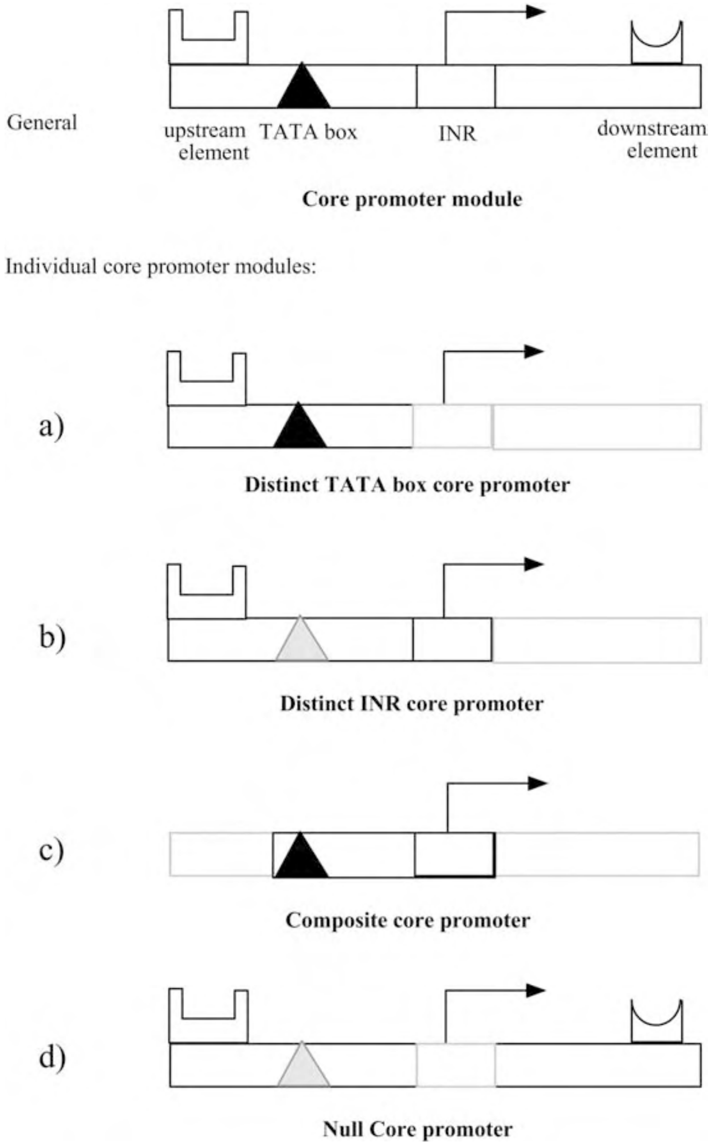


Figure 1 General structure of a Pol II core promoter and four different setups (a–d) of a Pol II core promoter. Simultaneous presence of all four elements is not always essential. The shapes above the bar symbolize additional protein-binding sites and the arrow indicates the TSS.

with expression from TATA-only promoters. This is also an example for of more indirect effects of specific arrangements in promoters that may not be apparent unless special conditions occur.

The last group consists of so-called null-promoters, which have neither a TATA box nor an initiator, and rely exclusively on upstream and downstream elements [66].

Basically, at least the four different core promoter types detailed above have been identified so far, all of which represent valid combinations of core promoter sites (reviewed in Ref. [66]). If the combinations involving upstream and downstream elements are also considered, seven core promoter modules are possible (most of which can be actually found in genes and consist of the four variants in Figure 1(a–d) adding upstream or downstream elements or both).

The only apparent common denominator of transcription initiation within a promoter would be that there must be at least one core promoter element anywhere within a certain region. This assumption is wrong. Both the spacing and/or sequential order of elements within the core promoter module are of utmost importance regardless of the presence or absence of individual elements (as a rule; however, there appear to be some exceptions). Moreover, many distinct promoters have requirements for specific upstream or downstream elements and will only function with their specific TF. Moving around the initiator, the TATA box and, to some extent, also upstream elements can have profound effects on promoter functions. For example, insertion of just a few nucleotides between the TATA box and an upstream TFBS (TF MyoD) in the desmin gene promoter cuts the expression levels by more than half [62]. Moreover, the promoter structure can affect later stages of gene expression like splicing [23]. It was also shown for the rat β-actin promoter that a few mutations around the TSS (i.e. within the initiator) could render that gene subject to translational control [7].

As a final note, the mere concept of one general TATA box and one general INR is an oversimplification. There are several clearly distinguishable TATA boxes in different promoter classes [35] and the same is true for the INR region, which also has several functionally distinct implementations as the glucocorticoid-responsive INR in the murine thymidine kinase gene [73], the C/EBP-binding INR in the hepatic growth factor gene promoter [48] or the YY-1-binding INR [90].

Most of the principles of variability and restrictions detailed above for the core promoter modules are also true for other promoter modules that modify transcriptional efficiency rather that determining the start point of transcription as the core promoter does. The bottom line is that the vast majority of alternative combinatorial arrangements of the elements that can be derived from a particular promoter might not contribute to the function of the promoter. Module-induced restrictions are not necessarily obvious from the primary sequences. Figure 2 shows a schematic Pol II promoter with the initiation complex assembled that illustrates that it matters where a specific

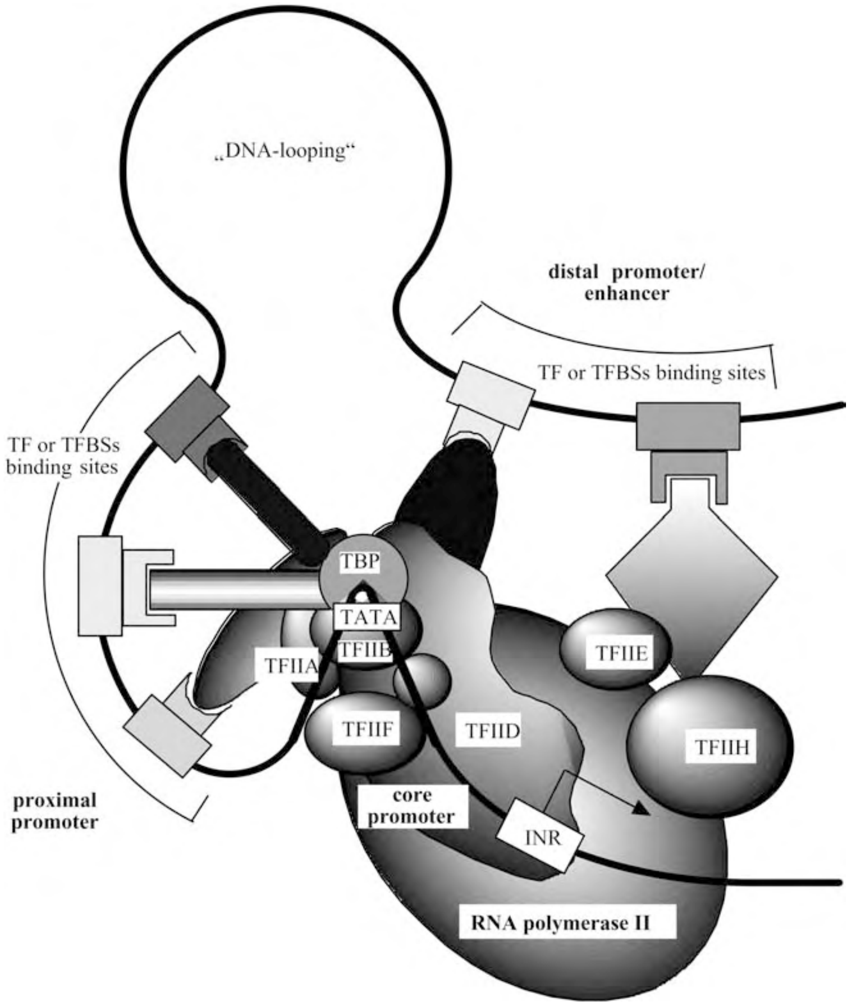


Figure 2 Transcription initiation complex bound to a schematic promoter.

protein is bound to the DNA in order to allow for proper assembly of the molecular jigsaw puzzle of the initiation complex. This is not immediately obvious from inspection of promoter sequences because there exist several (but a strictly limited set of) alternative solutions to the assembly problem. As complicated as Figure 2 may appear, it still ignores all aspects of chromatin rearrangements and nucleosomal positions, which also play an important role in transcription regulation. Stein and coworkers, initially in 1995 and in a 2001 follow-up paper, have detailed an example of the profound influence of these effects on promoter–protein complex assembly and function for the osteocalcin promoter [63, 84]. However, chromatin-related effects are not yet considered in any of the promoter prediction methods. Therefore, we do not go into any more details here.

1.7 Bioinformatics Models for the Analysis and Detection of Regulatory Regions

Algorithms used to analyze and detect regulatory regions are necessarily based on some kind of usually simplified model of what a regulatory region should look like. All of these models inevitably compromise between accuracy with respect to the biological model (the standard of truth) and computational feasibility of the model. For example, a computational model based on *a priori* 3-D structure prediction derived from molecular dynamics using sophisticated force fields may be the most accurate model for a region, but cannot be used for the analysis of real data due to excessive demand on computational resources. On the other hand, a model based on simple sequence similarities detected by IUPAC consensus (see also Section 2.1) sequences can be easily used on a PC, but results will usually not match the biological truth in an acceptable manner.

1.8 Statistical Models

It was noted several years ago that promoters and most likely also other regulatory regions like enhancers contain more TFBSs that nonregulatory sequences. Therefore, an analysis of the relative frequencies of such sites within a sliding window can yield some information on the potential regulatory character of a stretch of DNA, which is the prototype of simple statistical models. Several programs exist that rely to some extent on this type of statistics. Another set of statistical models calculates local GC content bias and uses this feature to discriminate potential promoters from other sequences. Such nucleotide bias statistics are only used in combination with other features (see Section 1.8.1) as they do not exhibit sufficient discriminatory power on their own.

A new breed of statistical models has been successfully introduced into promoter analysis more recently. These models focus on statistical analysis based on identification of promoter-associated words (not predefined such as TFBSs) using methods coming from other fields such as speech recognition. These methods are currently the best performers in promoter finding.

1.8.1 Mixed Models

It is clear from Section 1.6 that a binding site description-based pure statistical model is an oversimplification that will adversely affect the accuracy of prediction despite its attractive ease of implementation. Therefore, mixed models are also used that take at least some regional information into consideration and can be seen as statistical models split into compartments. Within the compartments solely statistical features are considered, but promoter organization is somewhat reflected by the arrangement of the compartments, which represent different promoter regions.

1.8.2 Organizational Models

The last category consists of models that try to closely follow the organizational principles of real regulatory regions. In order to accomplish this, individual promoter elements like TFBSs as well as their relative order and distances are encoded in a formal model, which reflects the setup of a single promoter or a small group of functionally similar promoters. Although they match the biological situation best, their widespread application requires an enormous amount of automation and background logistics such as high-quality promoter databases, automatic methods to derive the computational models as well as means of evaluating the resulting models. In the meantime, most of the basic requirements have been met, but real-life application is just picking up as this book is written.

However, such approaches are well suited for elucidating the molecular basis of coregulation of genes in a particular coexpressed cluster of genes from microarray experiments. So far this has been shown mainly for a simple eukaryote, *Saccharomyces cerevisiae* (yeast) [69]. Considerable progress has been made already in applying combinatorial TFBS models to higher eukaryotes such as mammalian systems, mostly based on experimental evidence [14, 40].

2 Methods for Element Detection

2.1 Detection of TFBSs

TFBSs are the most important elements within regulatory DNA regions like promoters or enhancers. The majority of the known TFs recognize short DNA stretches of about 10–15 nucleotides in length that show different degrees of internal variation. Successful detection of protein-binding sites in DNA sequences always relies on precompiled descriptions of individual binding sites. Such descriptions are usually derived from a training set of four or more authentic binding sites. However, the criteria applied for the decision whether a site is authentic or not vary considerably among authors of different publications. One of the first approaches to define protein-binding sites used IUPAC consensus sequences, which indicate the predominant nucleotide or nucleotide combination at each position in a set of example sequences (e.g.

SIGNAL SCAN [70]). The IUPAC string `TGASTCA` indicates that the first three positions are most frequently T, G and A, while the fourth position may be C or G, followed by T, C and A in most cases. IUPAC consensus sequences became very popular as they are extremely easy to define from even a small set of sequences, and their definition does not require more than a pencil and a sheet of paper.

However, IUPAC consensus sequences strongly depend on the sequence set used for definition because IUPAC consensus findings are based on majority rules. Adding or removing a single sequence can change the assigned nucleotide at a position while it would have little effect in a corresponding weight matrix. Cavener defined some rules that we have used for several years now and, in our experience, IUPAC consensus sequences defined that way can be useful [16]. However, IUPAC consensus sequences may reject biologically functional binding sites due to a single mismatch (or an ill-defined IUPAC sequence).

The concept of nucleotide weight matrix (NWM) descriptions was developed in the 1980s as an alternative to IUPAC strings (e.g. Refs. [83, 86]). Basically, weight matrices use an alignment of sequences to first generate a nucleotide distribution matrix representing the complete nucleotide distribution at each position of the alignment. Then some sort of weighting algorithm is used to adjust the matrix to the biological situation (also detailed in Section 5.3.1). However, although weight matrices proved to be generally superior to IUPAC strings, their greatest disadvantage is the absolute requirement for predefined matrices, which are more complicated to construct than IUPAC strings and require specific software. This delayed widespread use of weight matrices for almost a decade, although the methods were principally available. They remained mostly unused because only a few special matrices had been defined (e.g. Ref. [12]). The situation changed when in 1995 two (overlapping) matrix libraries for TF sites were compiled and became widely available almost simultaneously [17, 72]. MATRIX SEARCH [17] transformed the TRANSFAC database as completely as possible (starting at two binding sites for one factor) into matrices using a log-odds scoring approach. The MatInspector library [72] was originally largely based on a stringent selection from the matrix table of the TRANSFAC database, including the matrices derived from the ConsInspector library [32, 33] and several genuine matrices. The Information Matrix Database was compiled from the TRANSFAC matrix table and the TFD. In the meantime, the MatInspector library became independent from TRANSFAC and is updated regularly by Genomatix Software (Munich; currently more than 600 matrices), whereas IMD (another weight matrix database) has not been updated recently.

2.2 Detection of Novel TFBS Motifs

All the above covers the various approaches used to describe and find known TFBS motifs, i.e. there is always evidence that a known TF binds to such regions. There is another group of methods that join knowledge about evolutionary relationship of promoters with pattern-finding algorithms to detect phylogenetically conserved TFBSs. Examples of publications in this field include comparison of conserved human mouse patterns with [64] or without [61] direct sequence alignment, as well as approaches no longer restricted to two sequences such as FootPrinter [8] or PhyME, which includes overrepresentation into the probabilistic score of its findings [80].

A completely different set of methods deals with the detection of potential TFBS patterns solely based on their occurrence in a set of sequences without any biological knowledge about the particular TF binding to such regions. I separate such methods from the TFBS recognition methods as an unknown proportion of significant motifs detected this way may in fact not be TFBSs at all, but may be conserved for other reasons. Nevertheless, these methods do contribute to the generation of hypotheses about hitherto unknown TFBS patterns. Available matrix detection programs were reviewed some time ago [34] and a comparison of these methods by application to a test set of sequences has been published [36] (see Ref. [85] for a more recent review of the topic). A very recent study focused on matrix generation programs with no real emphasis on search programs [88]. For convenience, Table 1 summarizes some methods for the detection of TFBSs that are available in the internet with emphasis on programs featuring a WWW interface.

Various newer approaches have been published in the meantime, ranging from excellent purely mathematically motivated pattern detection (e.g. from Pevzner's group [50] or using self-organizing maps [65]) to strong connection

Table 1 Internet-accessible methods to detect promoter elements (TFBSs)

Program	Availability	Comments
MatInspector	http://www.genomatix.de	Genomatix matrices; free of charge use for academics (limited) after registration
SIGNAL SCAN	http://bimas.dcrt.nih.gov/molbio/ signal	IUPAC consensus library
MATRIX SEARCH	http://bimas.dcrt.nih.gov/molbio/ matrixs	IMD matrix library (TRANSFAC + TFD)
TFSearch	http://www.cbrc.jp/research/ db/TFSEARCH	TRANSFAC matrices
TESS	http://www.cbil.upenn.edu/tess/	TRANSFAC matrices
MATCH	http://www.gene-regulation.com/cgi-bin/ pub/programs/match/bin/match	TRANSFAC matrices; free of charge use for academics (restricted public version)

between biological [71] and experimental data with pattern detection [13]. This list only represents an arbitrary collection of very few papers in the field and the selection was purely driven by the desire to cite at least one method for each basic approach. I will not discuss *de novo* detection methods in any more detail here, as the major scope of this chapter is not *de novo* detection of patterns, but regulatory sequences analysis, which is usually based on precompiled pattern collections.

2.3 Detection of Structural Elements

Regulatory sequences are associated with a couple of other individual elements or sequence properties in addition to the factor-binding sites. Among these are secondary structure elements like the HIV-1 TAR region (*trans*-activating region, which constitutes an RNA enhancer, e.g. Ref. [10]), cruciform DNA structures (symmetric double hairpins of both strands in DNA, e.g. Ref. [92]) or simple direct repeats (e.g. Ref. [5]). Three-dimensional structures like curved DNA [54] also influence promoter function. Most of these elements can be detected by computer-assisted sequence analysis [20, 43], but none of them is really promoter specific and all such elements can be found frequently outside of promoters. The promoter or enhancer function arises from the combination of several elements that need to cooperate to exert transcription control which none of them can achieve alone. This also illustrates the main problem of promoter recognition. It is necessary to compile several individually weak signals into a composite signal, which then indicates a potential promoter without being overwhelmed by the combinatorial complexity of potential element combinations.

2.4 Assessment of Other Elements

Several methods employ statistical measures of sequence composition to include features of regulatory sequences, which cannot be described by the three types discussed above. These includes frequencies of oligonucleotides (dinucleotides, trinucleotides and hexamers are used most frequently), CpG islands (CG dinucleotides are usually underrepresented in mammalian genomes except in part of coding and regulatory sequences; CpG islands are regions where the dinucleotide is NOT underrepresented [38]) and periodicity of weak sequence patterns (`AA`, `TT`, etc.). Definitions of such elements are usually too weak to make any significant contribution to current prediction programs. However, this situation might well change due to the unprecedented amounts of continuous genomic sequences that become available in the course of the current genome-sequencing projects.

3 Analysis of Regulatory Regions

Basically, two different tasks can be distinguished in the analysis of regulatory regions. The first task is analysis aimed at the definition of common features based on sets of known regulatory sequences. This is a prerequisite for the definition of descriptions suitable for large-scale application for prediction of potential regulatory regions within new anonymous sequences, which can be seen as the second task.

3.1 Comparative Sequence Analysis

Comparative sequence analysis is one of the most powerful methods to deduce regulatory features and organization. Two main types of comparative analysis can be distinguished. The first approach compares regulatory regions, e.g. promoters within one species such as promoters coexpressed under particular conditions, or simply all (known) promoters within a genome to deduce general features. The second approach compares only orthologous regulatory sequences (again promoters are the most prominent representatives) in order to elucidate which features and elements have remained conserved in evolution. Such features should be closely associated with conserved functions of the corresponding regulatory regions. While comparative analysis within species affords no distinction between pure statistical findings and functional conservation, phylogenetic analysis of orthologous regulatory sequences should indicate predominantly functionally conserved features. However, intragenomic comparison may differentiate between individual functions, whereas phylogenetic analysis will always yield a summary over all conserved functions. Thus, very often a combination of both approaches is the best way to go [25].

3.2 Training Set Selection

One of the most important steps in comparative sequence analysis is the selection of suitable training sets of sequences. If a training set of promoters consists only of constitutively expressed sequences (constant level of expression, no or little regulation), little can be learned about any kind of tissue-specific expression regardless of the methods applied. Inclusion of too many wrong sequences (e.g. that are not promoters at all or promoters not involved in the regulation under investigation, see alternative promoters below) may also prevent any meaningful analysis. Although this observation appears trivial at first, it becomes a real issue when data are scarce and less well-characterized sequences have to be used.

Control sets known not to be functionally similar to the training sets are about as important as the training sets. However, true negative regions are even scarcer than known regulatory regions. Negative often means just "no positive functions found", which can also be due to failures or simply means that the sequences have not been tested at all. Therefore, statistical negative control sequences are often required. Random sequences can be generated easily, but often are of limited use, as they do not represent several important features of natural DNA correctly. This includes underrepresented features (e.g. CpG islands), asymmetric features (e.g. strand specificity), local changes in GC content or repetitive DNA elements. Selection of appropriate control sequences can be a major effort, but is also crucial for the validity of the evaluation of any method. Common problems with controls are either known or unknown biases in the control set or circularity problems, i.e. the training and the test sets of sequences are related or overlap. The availability of large continuous stretches of genomic DNA from the genome-sequencing projects constantly improves this situation. Genomic sequences should always be the first choice for controls as they reflect the natural situation.

3.3 Statistical and Biological Significance

The quality of sequence pattern recognition is often optimized to improve the correlation of the methods with the data (positive and negative training sets). However, in most cases it is not possible to collect sufficient data to perform a rigorous correlation analysis. Therefore, bioinformatics methods often rely on statistical analysis of their training sequences and optimize for the statistically most significant features. Unfortunately, this kind of optimization does not always reflect the evolutionary optimization of regulatory sequences that is always optimizing several features at once. This problem is different from overfitting of data as it is more about optimization criteria than parameter fitting *per se*.

The dynamics of biological function often necessitates suboptimal solutions. For example, real sequences usually do not contain binding sites with the highest affinity for their cognate protein because binding *and* dissociation of the protein is required for proper function. The perfect binding site with the highest binding affinity would interfere with the dissociation and is therefore strongly selected against.

3.4 Context Dependency

The biological significance of any sequence element is defined by the regulatory function it can elicit. This is usually dependent on a functional context rather than being a property of individual elements. Therefore, statistical significance of the features or scores of individual elements is neither necessary nor sufficient to indicate biological significance. Recognition of the functional context in an essentially linear molecule like DNA can be achieved by correlation analysis of individual elements, which became an important part of all semi-statistical or specific modeling approaches discussed below. The context is also an important parameter in statistical analysis. For example, an element frequently found all over the genome could become even statistically significant if only the immediate vicinity of a binding partner's binding sites is analyzed such as in case of transcriptional modules. Therefore, lack of statistical significance may just indicate that the wrong context was chosen for the analysis.

4 Methods for Detection of Regulatory Regions

There are several methods available for the prediction of regulatory DNA regions in new sequence data. Table 2 lists methods available with a special focus on programs that provide a WWW interface. Unfortunately, there is no "one-does-it-all" method, and all methods have their individual strong and weak points. There was a fairly recent review on the subject including most relevant programs, with one exception [4]. The program PromoterInspector

Table 2 Internet-accessible promoter/promoter region prediction tools

Program	Availability	Comments
Promoter prediction *Ab initio* promoter finding (large-scale sequences)		
PromoterInspector	http://genomatix.de	free of charge use for academics (limited) after registration
Dragon PromoterFinder	http://research.i2r.a-star.edu.sg/ promoter/promoter1_5/DPF	free for academics
Promoter finding in preselected sequence ranges		
Eponine	http://servlet.sanger.ac.uk:8080/ eponine	free for academics
FirstEF	http://rulai.cshl.org/tools/FirstEF	free for academics
Promoter module/region recognition		
ModelInspector	http://www.genomatix.de	free of charge use for academics (limited) after registration; modules of two TF sites (MatInspector library)

[77] was not included as it predicts promoter regions and neither strand orientation nor the TSS.

A program doing an excellent job in one case might be a complete failure in another case in which other methods are successful. Therefore, we will describe a number of methods without intending any rank by order of discussion. We will rather follow the functional hierarchy that appears to apply to the different regulatory regions. However, the apparent best application range will be indicated.

4.1 Scaffold/Matrix Attachment Regions (S/MARs)

A chromatin loop is the region of chromosomal DNA located between two contact points of the DNA with the nuclear matrix marked by so-called S/MARs. The nuclear matrix is a mesh of proteins filling the interdomain space inside the nucleus where S/MARs form highly flexible structures that are necessary, but not sufficient, for anchoring at chromosomal DNA to the matrix [42].

Transcriptional regulation requires the association of DNA with this nuclear matrix, which retains a variety of regulatory proteins. S/MARs are composed of several elements, including TFBSs, AT-rich stretches, potential cruciform DNA and DNA-unwinding regions, to name a few of the most important S/MAR elements. There is an excellent recent review on chromatin domains and S/MAR functions [9]. Singh and coworkers published a method to detect potential S/MAR elements in sequences and made the method available via WWW (http://www.ncgr.org/MAR-search/) [79]. Their method is based on a statistical compilation of the occurrence of a variety of S/MAR features (called rules). Accumulation of sufficient matches to these rules will be predicted as potential S/MAR regions. The specificity of the method depends critically on the sequence context of the potential S/MAR sequences. Another approach utilizes a single S/MAR associated sequence element to locate potential S/MARs [91]. Therefore, results are difficult to evaluate by comparisons. We developed another approach to define especially AT-rich MARs called SMARTest, which is available on the web at http://www.genomatix.de. SMARTest is based on a library of MAR-associated nucleotide weight matrices and determines S/MARs independent of any larger sequence context [37]. Therefore, the method is suitable for testing isolated S/MAR fragments. MARFinder and SMARTest are complementary, and should be seen in combination rather than as alternatives.

4.2 Enhancers/Silencers

Enhancers are regulatory regions that can significantly boost the level of transcription from a responsive promoter regardless of their orientation and distance with respect to the promoter as long as they are located within the same chromatin loop. Silencers are basically identical to enhancers and follow the same requirements, but exert a negative effect on promoter activities. Both regulatory regions are also relevant in disease processes, as detailed in a recent review [55]. At present there are no specific programs to detect enhancers and silencers. However, programs designed to detect the internal organization of promoters are probably also suitable to detect at least some enhancers and silencers since these regions often also show a similar internal organization as promoters.

4.3 Promoters

Promoters were described in detail above – they are just mentioned here again to place them into context.

4.4 Programs for Recognition of Regulatory Sequences

There are several ways promoter recognition tools can be categorized. We will focus on the main principles and intended usage of the programs rather than technical details. Two generally distinct approaches have been used so far in order to achieve *in silico* promoter recognition. The majority of programs focus on *general promoter recognition*, which represents the first category.

The second category of tools aims at *specific promoter recognition* relying on more detailed features of promoter subsets like combinations of individual elements. The beauty of this approach is its excellent specificity, which is extremely helpful if only promoters of a certain class are of interest or megabases of sequences have to be analyzed. The bad news here is limited applicability, i.e. each promoter group or class requires a specifically predefined model before sequences can be analyzed for these promoters. This may result in a huge number of false negatives in large-scale analysis.

We will briefly discuss individual methods in these two categories with emphasis on the implementation of the biological principles of promoter features. Recently, a practical comparison of the majority of available tools based on general promoter models has been carried out [4], which was the first large-scale update since the original comparison carried out by Fickett and Hatzigeorgiou in 1997 [29]. There was another review in between those two studies by Ohler and Niemann [67]. Therefore, we will not go into details on the performance of the methods.

4.4.1 Programs Based on Statistical Models (General Promoter Prediction)

These programs aim at the detection of Pol II promoters by a precompiled general promoter model that is part of the method. Learning methods range from supervised artificial neuronal networks over statistical analyses to simple counting of features to a threshold. One group of programs in this category (see below) concentrates on recognition of core promoter properties and infers promoter location solely on that basis, whereas the other group consists of programs that take into account also the proximal promoter region of about 250–300 nucleotides upstream of the TSS. General recognition models were usually based on training sets derived from the Eukaryotic Promoter Database (EPD) and various sets of sequences without known promoter activities. The EPD originally was an excellent collection of DNA sequences that fulfill two conditions: they have been shown experimentally to function as promoters and the TSS is known. Recently, EPD also started to incorporate promoters not fulfilling these stringent conditions [78].

The beauty of the above approaches is their generality, which does not require any specific knowledge about a particular promoter in order to make a prediction. This appears ideal for the analysis of anonymous sequences for which no *a priori* knowledge is available. The bad news was that the specificity of all such general approaches implemented was very limited for quite some time. However, the development of PromoterInspector [77] heralded a new era of promoter prediction, combining acceptable sensitivity with high specificity. Other programs that followed performed comparably [3]. These general promoter prediction approaches were the first to provide acceptable *a priori* promoter prediction on a whole chromosome and now genome scale [76]. Specificities were originally reported just below 50%, but in the meantime many of the orphan predictions (in the middle of unannotated sequence) have found their genes and transcripts boosting specificity to between 80 and 90%. Only a really complete annotation of the genomes will tell the true specificity of those methods. Nevertheless, it is clear that the goal of highly specific promoter prediction in whole mammalian genomes has been achieved.

Some general promoter model-based programs employ methods already described for identification of individual promoter elements (usually TBFBS IUPAC or weight matrix descriptions), but try to derive more general features from a collection of such elements rather than emphasizing individual elements. These methods may be called *statistical element analyses* and treat the proximal promoter as a purely statistical problem of TFBS accumulations, sometimes fine-tuned by some sort of weighting based on occurrence frequencies of TFBSs in promoters as compared to a negative sequence set. Despite the complicated modular structure of promoters outlined above there is a solid rational basis for this general model. All promoters must have a functional core promoter module often containing a TATA box, which is the prime target of the majority of the general promoter prediction tools. This is also one of the reasons that some programs confine their analysis to the core promoter region, which avoids problems with the much more diverse proximal regions. Biological knowledge is solely used to select the training sets and a variety of methods is used to learn the distinctive patterns.

Without exception, TFBS-based statistical element analysis suffers from a huge number of false-positive predictions (typically about one prediction in 10 000–30 000 nucleotides).

4.4.2 Programs Utilizing Mixed Models

These programs also rely on statistical promoter models, but include directly or indirectly some organizational features of promoters, placing them in between the pure statistical models and attempts to approximate the biologically important structured organization of promoters. Again, the first-generation methods will only be summarized. FunSiteP [53] as well as the approach taken by Audic and Claverie [2] fall into this category.

4.4.3 Programs Based on Specific Promoter Recognition

The second, more recent and far more successful concept should be called *functional element analysis*, as it relies heavily on biological knowledge about the relative importance of individual elements and derives discriminative features on that basis. These methods carry out a sophisticated compositional analysis of the proximal promoter analysis to detect unique features within that region that can be used to distinguish promoters from nonpromoters without understanding the details, but using any pre-existing knowledge for feature selection.

This category of methods introduces the functional context in the form of heuristic rules or tries to learn the context from comparative sequence analysis. These methods emphasize specific modeling of promoters or promoter substructures rather than general recognition. Therefore, it is not possible to directly assess the promoter prediction capabilities of these methods. However, in many cases recognizing a common substructure between promoters can be very helpful, especially for experimental design. Although these programs were also published during the time the first-generation general promoter prediction programs appeared, they are still useful in whole-genome scans due to their very high specificity, warranting a more detailed discussion here.

The method FastM was derived from the program ModelGenerator [31] and takes advantage of the existence of NWM libraries. It can be accessed via a WWW interface (http://genomatix.gsf.de part of GEMS launcher) and allows for a straightforward definition of any modules of two TFBSs by simple selection from the MatInspector Library [72]. This now enables definition and detection of wide variety of synergistic TFBS pairs. These pairs are often functional promoter modules conferring a specific transcriptional function to a promoter as shown in Refs. [52, 56]. FastM models of two binding sites can successfully identify promoters sharing such composite elements, but are not promoter specific. Composite elements can also be located in enhancers or similar structures. The latest version of FastM enables definition of complete, highly specific promoter class models including up to 10 individual elements, also including IUPAC strings, repeats and hairpin structures.

The program FrameWorker [15] automates several of the steps taken manually in FastM in order to ground specific promoter modeling on as much an algorithmic basis as possible. FastM requires crucial parameters such as strand orientation, distance ranges, order of elements, as well as the individual nature of the elements (e.g. which weight matrix to use) to be determined by the user. FrameWorker, in contrast, automatically determines theses parameters from a comparison of an (still manually selected) set of input sequences within user-defined ranges. However, determination of the individual weight matrices to be used, as well as their number, distances and relative order, does not require previous knowledge.

Another approach aimed at modeling promoter substructures consisting of two distinct elements is TargetFinder [58]. This method combines TFBSs with features extracted from the annotation of a database sequence to afford selective identification of sequences containing both features within a defined length. The advantage is that TargetFinder basically also follows the module-based philosophy, but allows inclusion of features that have been annotated by experimental work for which no search algorithm exists. Naturally, this excludes analysis of new anonymous sequences. The program is accessible via a WWW interface (http://gcg.tigem.it/TargetFinder.html).

It should be mentioned here that Fickett also employed the idea of a two-TFBS module to successfully detect a subclass of muscle-specific regulatory sequences governed by a combination of MEF2 and MyoD [28]. However, this was also a very specific approach and no general tool resulted from that work. The MEF2/MyoD model can be used to define a corresponding module with FastM. Wasserman and Fickett also published a modeling approach based on clustering of a preselected set of NWM (defined in the same study) correlated with muscle-specific gene expression [93]. They were able to detect about 25% of the muscle-specific regulatory regions in sequences outside their training set and more than 60% in their training set. They classify their method as regulatory module detection. However, their results suggest that they probably detect a collection of different, more specific modules with respect to the definition given above. Although the method is not promoter specific and the specificity is moderate, it is a very interesting approach that has potential

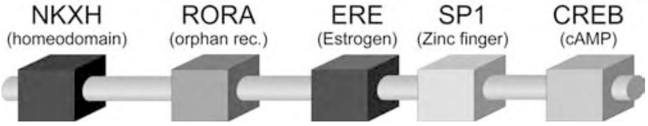


Figure 3 GFAP promoter model conserved in human mouse and rat promoter. The boxes indicate the individual TFBSs found and the bar indicates the genomic DNA.

for further development, as also became evident from follow-up publications of the same authors [30,94].

Generally, this group of methods achieves much higher specificity than the first-generation programs following general models. However, the price for this increase in specificity is usually restriction of the promoter models to a small subset (class) of promoters.

The model of the glial fibrillary acidic protein (GFAP) promoter shown in Figure 3 was derived from a comparison of the human, mouse and rat GFAP promoters. This model contains five different TFBSs and was derived from the set of three sequences using GEMS Launcher (Genomatix). This model recognizes a single sequence, the GFAP promoter, when searched against more than 36 000 human promoter sequences and thus is absolutely gene specific. Interestingly, if the search is carried out with relaxed stringency (allowing for less-perfect matches) only a second sequence comes up, the DGAT2 gene, which is the diacylglycerol *O*-acetyltransferase homolog 2 (homolog to mouse). From the literature it becomes immediately evident that both genes are brain-expressed (GFAP is brain/astrocyte specific) and both are genes associated with insulin signaling. Thus the promoter model-based search found biologically linked genes.

4.4.4 **Early Attempts at Promoter Prediction**

There are various programs that might be called first-generation programs for promoter prediction, some of which were absolutely instrumental in paving the way towards the newer developments, but are no longer of practical use. For that reason they will only be summarized here and not discussed in detail. The first exception to this rule will be Promoter Scan, as this was really the first program ever published for promoter prediction in mammalian sequences and served as a role model for a number of other developments.

Several of the general promoter prediction programs followed the basic design of Prestridge who used the EPD by Bucher's group [78] to train his software for promoter recognition. His program Promoter Scan was the first published method to tackle this problem [70]. He utilized primate nonpromoter sequences from GenBank as a negative training set and included the proximal promoter region in the prediction. The program uses individual profiles for the TFBSs indicative of their relative frequency in promoters to accumulate scores for DNA sequences analyzed. Promoter Scan employs the SIGNAL SCAN IUPAC library of TFBSs [70], introducing a good deal of biological knowledge into the method, although modular organization of the proximal region is necessarily ignored. Results of the first version were combined with the Bucher NWM for the TATA box, which served as a representation of the core promoter module [12].

Other methods following a similar design will not be discussed in detail, but should be mentioned. These include PromFD by Chen and coworkers [18], and the programs TSSG/TSSW from Solovyev's group, which are basically gene prediction methods that include promoter prediction [82]. Other programs in that category are XLandscape [60] and PromFind [44]. Michael Zhang published a new method to detect TATA-box containing core promoters by discrimination analysis.

5 Annotation of Large Genomic Sequences

Many of the methods discussed above were developed before the databases started to be filled with sequence contigs exceeding 100 000 nucleotides in length. The complete human genome draft now contains more than 3 billion nucleotides and many more genomic sequences of similar size are entering the databases. This changes the paradigm for sequence annotation. While complete annotation remains an important goal, specific annotation becomes mandatory when even individual sequences exceed the capabilities of researchers for manual inspection. Annotation of genomic sequences has to be fully automatic in order to keep pace with the rate of generation of new sequences. Simultaneously, annotations are embedded into a large natural context rather than residing within relatively short isolated stretches of DNA. This has several quite important consequences.

5.1 Balance between Sensitivity and Specificity

We will confine the discussion here to regulatory regions, but the problems are general. A very sensitive approach will minimize the amount of false-negative predictions and thus is oriented towards a complete annotation. However, this inevitably requires accepting large numbers of false-positive hits, which easily outnumber the true-positive predictions by an order of magnitude.

In order to avoid this problem methods can be designed to yield the utmost specificity (e.g. specific promoter modeling as discussed above). Here, the catch is inevitably a high number of false-negative results, which also may obscure 70–90% of the true-positive regions. The newer developments of general, but still specific, promoter finding (especially Refs. [3, 77]) may provide a way out of the dilemma. Once a rough annotation has been achieved, other methods can come in to locate promoters reliably in more restricted search spaces such as the FirstExon Finder [24] and Eponym [26]. There was a recent survey of promoter finding in genomic sequences emphasizing that suitability of methods for analysis of large genomic sequences cannot be inferred from limited tests with short samples, which could be referred to analysis in a "sheltered environment" [4].

Gene (or gene group)-specific methods were shown to produce more that 50% true-positive matches in their total output (e.g. Ref. [35]), but recognize just a small fraction of all promoters, which is inevitable for a function-specific model. A single specific model like the phylogenetically conserved GFAP promoter model (Figure 3) matched only once in the human genome, indicating that it is absolutely specific for the GFAP gene.

Definition of the required number of specific models based on current technologies was not a feasible task until recently. However, new developments have already been initiated to overcome the current obstacles and Genomatix is actually working on a genome-wide library of evolutionarily conserved organizational promoter models.

It is quite evident that functional promoter analysis in laboratories is capable of dealing specifically with several hundred or even thousand predicted regions, whereas predicting several hundred thousand or even millions of regions remains out of reach. However, recent improvements of laboratory high-throughput technologies such as location of the TSS by the so-called oligo-capping method [87] have provided an unprecedented amount of verified TSSs (which by definition are located within the promoters). Nevertheless, enhancements of the specificity of promoter recognition *in silico* will also be required as the oligo-capping method has an inherent error rate of 20–30%. Both developments will meet sometime in the future to close the gap in our knowledge about the location of promoters in the genome. More elaborative approaches will be required both in the laboratory as well as in bioinformatics in order to also understand the functionality hidden within these regulatory sequences. It is clear from the past and present developments that bioinformatics will probably cover significantly more than half of that path.

5.2 Genes – Transcripts – Promoters

Originally, the notion was that one gene would represent one function. We learned in the early days of molecular biology during the 1980s that this is not quite true and that one gene may very well have several functions. However, it did not become clear how this is realized until the large-scale

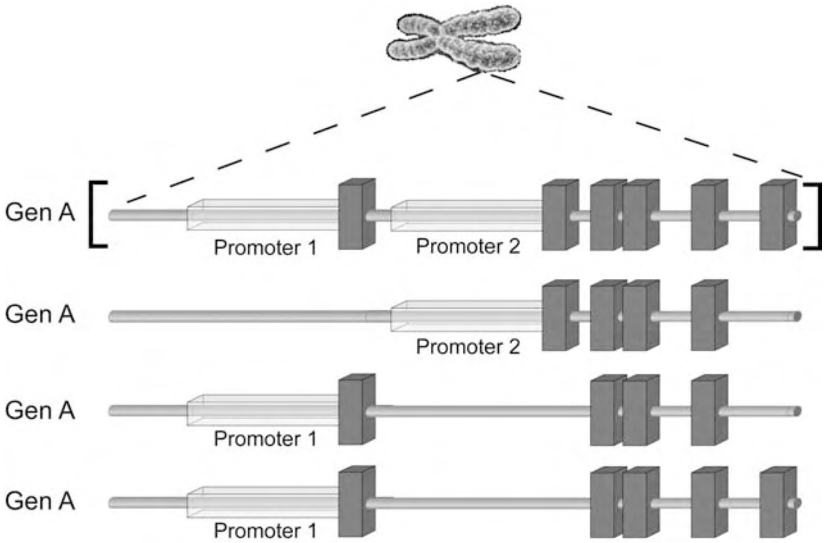


Figure 4 Genomic organization of genes, promoters and transcripts. The transparent boxes indicate promoters, the grey boxes indicate exons and the grey bars indicate the genomic sequence. The brackets delineate the locus of the gene.

mapping and sequencing effort provided us with a better insight into genomic organization. This knowledge has changed our perception of a gene. A gene is no longer an entity, but rather a container with individual transcripts representing the entities. Figure 4 illustrates this new notion schematically. The area in brackets indicates the genomic locus of the gene. This region can be larger than a million base pairs in some cases, providing the space for the complex inner organization. The line with the brackets indicates the genomic structure, such that both promoters and all exons are in a linear arrangement with no clue about the functional links between the individual elements. The lines below refer to individual transcripts, with two transcripts exhibiting alternative splicing originating from promoter 1, while another transcript originates from promoter 2. The important consequence is that this gene may behave like two independent genes with respect to regulation, and the transcript originating from promoter 2 can be completely independent in terms of regulation and function from the other transcripts. They may even encode quite different proteins.

From this it is immediately evident why the paradox of humans having only a moderate amount of genes in excess to *Drosophila* or *Caenorhabditis elegans* is not a real paradox. The inner complexity of transcript and regulatory combinations more than compensates for the apparent lack in total gene numbers. If we count transcripts rather than genes, mammalians do have close to or even above 100 000 – a number earlier pondered for genes as required for the observed complexity. It just turns out that within regulatory sequences as well as within the whole genome, complex hierarchical organization prevails over simple numbers of elements. This is not surprising as the hierarchical principle allows a much more economic utilization of genomic sequences.

5.3 Sources for Finding Alternative Transcripts and Promoters

Of course, once we realized that alternative transcripts as well as alternative promoters are important in general, the question arises how to cope with this extra level of complexity. There are several consequences that need to be taken into account. First, many expressed sequence tags (ESTs) sequences so far simply dismissed as "genomic contamination" may in fact indicate alternative transcripts, as what is an exon in one transcript can be an intron in another. The same is true for a predicted or experimentally verified promoter. If such a promoter was located inside a well-known gene, it was readily dismissed as a false positive, because we already "knew" that the promoter was further upstream. We have seen many cases in which the "false" promoter has found its own transcript in the meantime and was promoted from false to alternative. However, this has blurred the line between "true" and "false" considerably. What is apparently true for one condition (e.g. in one tissue) may be "false" for another condition (e.g. in another tissue). This dilemma is far from being finally solved, but as a practical approach we have adopted a policy of "multiple-evidence" support. The idea is very simple – both theoretical as well as laboratory-based approaches may yield false results. However, if two or more *independent* methods suggest the same conclusion, it is much more likely to be true than that both methods made exactly the same mistake. For example, if oligo-capping indicates a TSS, which happens to be located right inside a predicted promoter, we take this as evidence for a real promoter. Both methods are totally independent of each other and the chance of a result converging by chance is absolutely minimal. Based on this concept, ElDorado (Genomatix) has accumulated more than 150 000 primary transcripts as well as promoters for five mammalian species so far and we are quite confident that we have not yet seen the end of the story.

5.4 Comparative Genomics of Promoters

We have alluded to the "multiple-evidence" approach already in the previous section. However, there is very powerful line of evidence that has not yet been mentioned – the evolutionary conservation of gene regulation. This is one of the most direct lines of evidence towards the functional conservation of promoters as functional regions or elements are far better conserved that the sequence in general. We took advantage of this fact and developed a complete strategy affording the identification and subsequent mapping and analysis of orthologous promoters (Genomatix, patent pending). On top of identification of promoters of orthologous genes, this also includes finding the individual promoters within each species that correspond to each other, which we termed orthologous promoter sets. This is very important for subsequent analysis as functional elements, because functional element conservation is only detectable within orthologous promoter sets. Of course, this approach becomes stronger and stronger as the number of available genomes rises. As of 2004, this enabled us to detect or confirm more than 10 000 promoters in the human genome, making comparative genomics of promoters a major source of promoter annotation (as taken from the ElDorado statistics; Genomatix).

6 Genome-wide Analysis of Transcription Control

If the focus is broadened from individual genes or small gene groups towards looking at the whole genome it is no longer sufficient to just take promoters into consideration. On a genome-wide scale the hierarchy of gene regulation comes into the picture in full force. First, expression of genes on the mRNA level by transcription requires the locus of the gene to be accessible. Regulation of gene expression at the DNA level effected not by TFs, but by other factors elsewhere in the genome, is generally termed epigenetic regulation. This includes regulation by alternation of the chromatin structure, where DNA and histone modifications (e.g. DNA methylation or histone acetylation) play a role and the S/MAR elements discussed above become important. Whether the chromatin structure is open or closed determines whether a promoter becomes available for transcription or not. Thus, a gene with the perfect setup within its promoter(s) can be silent even if all the required TFs are present, provided the chromatin is closed, thus blocking access of these factors to the promoter. Let us assume that the chromatin is in an open, i.e. accessible, state. Even this does no guarantee active transcription of the embedded genes. Local DNA methylation can interfere, an active silencer can specifically block individual genes or one crucial factor may be missing or sequestered (e.g. the nuclear factor NFκB can be blocked by its inhibitor IκB, rendering it nonfunctional the despite presence of the protein). Active transcription is only observed when all conditions are right: the chromatin is open, no repressor is active, and all crucial factors can actually access their respective binding sites on the promoter and enhancer, if one is required. There is also a very old mechanism that seems to gain importance in the regulation of gene expression again – antisense transcription [98]. This means that the same region is transcribed in both directions, resulting in complementary RNAs that can form dimers and thus cancel each other out, as RNA dimers are prone to be destructed immediately. This very complex situation is a formidable safeguard against spurious expression of genes, which could be disastrous for a cell.

6.1 Context-specific Transcripts and Pathways

The many conditions that have to be met to enable the expression of a gene are also behind the differential expression of individual transcripts often coupled to particular pathways. Transcripts can be cell/tissue specific, pathway specific (or better associated as complete specificity is rare) or tied to a particular developmental stage of an organism. This emphasizes the important fact that biological function is tied to the transcript/protein, not to the gene, which may well encode various functions in various transcripts. There is also an important consequence for the analysis of regulatory networks behind signaling or metabolic pathways. It is not sufficient to identify which genes are involved in that pathway, but of utmost importance to identify the promoters associated with that particular transcript/pathway. This is also the reason that the very same pathway containing the same genes can still be differently regulated in different tissues, if different transcripts/promoters are involved in the different tissues. The upside of this complicated situation is that regulatory analysis based on the correct promoters is as close to the real biological situation as we currently can get with *in silico* methods. As it does not make any sense to simplify biology to fit our generic models it is well worth the effort to identify the conditionally important transcripts and promoters as this assures biological importance of the results.

6.2 Consequences for Microarray Analysis

Another field to which the bioinformatics of regulatory DNA regions can be expected to contribute significantly is the analysis of results from high-throughput experiments in expression analysis (e.g. all forms of expression arrays). Due to the discontinuous nature of regulatory regions there is no way of deducing common regulatory features from the expression data directly which are usually based on coding regions. However, the general availability of the corresponding genomic regulatory regions for many (and very soon all) of the genes analyzed in an expression array experiment enables attempts to elucidate the genomic structures underlying common expression patterns of genes. Expression arrays (described in detail in Chapters 24–28) directly deliver information, *which* genes are expressed *where* under the conditions tested. However, they cannot provide any clue to *why* this happens or how the same genes would behave under yet untested conditions. Identification of functional features by comparative sequence analysis (e.g. promoter modules) can reveal different functional subgroups of promoters despite common regulation under specific conditions. Consequently, the detection of known functional modules can suggest expression patterns under yet untested conditions [95]. Moreover, the organizational structures of promoters can also be used to identify additional potential target genes either within the same organism in other genomes or via comparative genomics. Given the exponential number of possibilities for combinations of conditions, bioinformatics of regulatory sequences will also become instrumental for the rational design of expression arrays as well as for selection of experimental conditions.

While this basic conduct of analysis of microarray data remains unchanged, our growing knowledge of alternative transcripts and alternative promoters has far-reaching consequences on strategies employed to analyze transcript levels on a large scale – the microarrays of DNA chips. The most obvious consequences of course are for the analysis of microarray data based on current chip designs that can be purchased from several vendors. As this is the most clear-cut consequence, let us focus on this point first. If there is a single transcript from a single promoter for a given gene, there is no problem, as none of the above complications applies. However, according to current knowledge probably more than 80% of all genes have alternative transcripts and maybe more than half also alternative promoters [99]. Both numbers are rough estimates from what we already know and can be expected to rise even further. This illustrates nicely that the carefree situation of single transcripts and promoters is most likely the exception, not the rule, for genes represented on micorarrays. It has already been recognized that this may cause problems with the traditional way of probe selection, rendering part of the probes on a microarray uninformative [39]. The problem with alternative splicing has been recognized already and studies in that direction have been carried out [49]. There are also efforts under way by microarray manufacturers to take alternative splicing into account. Fortunately, it became possible to check which probes can be reliably used and which probes might cause problems thanks to the high-quality genomic sequences available and our increasing knowledge about alternative transcripts. It should be noted that the set of useful probes depends to some extent on the experimental conditions, not the array used. Some probes might be very informative, whereas the alternative transcripts also recognized by the same probes are not expressed. Use of such probes might cause problems under conditions in which such alternative transcripts are coexpressed.

However, the case of alternative promoters is much less well recognized, but is of equal importance as in many cases transcripts appear to be the same, but originate from different promoters. For example, the CYP19A (also known as aromatase) gene that has at least seven promoters (probably even 10), all of which appear to encode the same transcript. The reason for that paradox is that all promoters are linked to alternative noncoding first exons of almost identical length all of which splice invariably to the identical coding region comprised of nine additional exons. Thus, basically all probes recognize any of the transcripts indifferently. However, events important in breast cancer include a switch of promoter usage not detectable that way [19]. Only transcript-specific probes will help here and they can only be designed based on knowledge about the alternative promoters.

As already mentioned, it is possible to reduce the amount of potentially ambiguous probes by utilizing the existing knowledge of alternative transcript structures [49]. Based on the huge promoter collection in ElDorado, Genomatix is currently evaluating genome-wide probe sets that are specific for alternative promoter usage in order to afford the design of microarrays that will directly indicate promoter selection. This will be of great use for subsequent promoter analyzes as it will take the guesswork out of the selection of promoters. This will also be the only way to tackle the problems of closely related transcripts such as in the case of the CYP19A gene discussed above. It is safe to assume that transcript- and promoter-specific microarrays will become the standard in the near future, bringing the results obtained with such arrays a lot closer to the underlying molecular mechanisms that present-day arrays allow.

7 Conclusions

The experimental dissection of functional mechanisms of transcription control has gained an enormous momentum over recent years. The ever-increasing number of publications on this topic bears witness to this development, which found one early hallmark manifestation in the introduction of a new section in the *Journal of Molecular and Cellular Biology* entirely devoted to analysis of transcription control, which just spearheaded widespread publication of similar articles in most other leading journals. The complex interleaved networks of transcription control certainly represent one of the cornerstones on which to build our understanding of how life functions, in terms of embryonic development, tissue differentiation, and maintenance of the shape and fitness of adult organisms throughout life (see also Chapter 21). This is also the reason why both the experimental analysis and the bioinformatics of transcription control will move more and more into the focus of medical/pharmaceutical research. A considerable number of diseases are directly or indirectly connected to alterations in cellular transcription programs (e.g. most forms of cancer). We recently demonstrated how promoter analysis can be used to elucidate some underlying molecular networks in insulin signaling with relevance to the maturity onset of diabetes of the young (MODY [25]). Furthermore, many drugs influence transcription control via signaling pathways (triggering TFs) [47], which could also be connected to certain side-effects of drugs [73]. The various genome-sequencing projects will provide us with a complete catalog of the components of a number of mammalian species probably within a few years. This will complement the blueprint of the material basis of a human already derived from the human genome sequence. However, only the analysis of the regulatory part of the genome and the corresponding expression patterns and the complex metabolic networks will provide deeper insight into how the complex machinery called life actually works. Definition and detection of regulatory regions by bioinformatics will contribute to this part of the task, and will become instrumental in guiding experimental approaches as well.

As a final note it should be emphasized that transcriptional regulation necessarily involves thousands of proteins, which is why proteomics analyses will also make important contributions to our understanding of regulatory events (see Chapter 28). However, despite its much longer history, protein research has not yet reached the level where it can be readily merged with the DNA-based analysis of transcription control. Nevertheless, we are quite confident that in the very near future protein research will be as integrated into the analysis of genome regulation as are nucleotide sequence- based methods today. Biology simply cannot be divided into DNA, RNA and protein "fields" as all of this is required to define and support the wonderful concerted action called life.

References

1 Arnone, M. I. and E. H. Davidson. 1997. The hardwiring of development: organization and function of genomic regulatory systems. Development **124**: 1851–64.

2 Audic, S. and J. M. Claverie. 1997. Detection of eukaryotic promoters using Markov transition matrices. Comput. Chem. **21**: 223–7.

3 Bajic, V. B., S. H. Seah, A. Chong, G. Zhang, J. L. Koh and V. Brusic. 2002. Dragon PromoterFinder: recognition of vertebrate RNA polymerase II promoters. Bioinformatics **18**: 198–9.

4 Bajic, V. B., S. L. Tan, Y. Suzuki and S. Sugano. 2004. Promoter prediction analysis on the whole human genome. Nat. Biotechnol. **22**: 1467–73.

5 Bell, P. J., V. J. Higgins, I. W. Dawes and P. H. Kissinger. 1997. Tandemly repeated 147 bp elements cause structural and functional variation in divergent MAL promoters of *Saccharomyces cerevisiae*. Yeast **13**: 1135–44.

6 Bergers, G., P. Graninger, S. Braselmann, C. Wrighton and M. Busslinger. 1995. Transcriptional activation of the *fra-1* gene by AP-1 is mediated by regulatory sequences in the first intron. Mol. Cell. Biol. **15**: 3748–58.

7 Biberman, Y. and O. Meyuhas. 1997. Substitution of just five nucleotides at and around the transcription start site of rat beta-actin promoter is sufficient to render the resulting transcript a subject for translational control. FEBS Lett. **405**: 333–6.

8 Blanchette, M. and M. Tompa. 2003. FootPrinter: a program designed for phylogenetic footprinting. Nucleic Acids Res. **31**: 3840–2.

9 Bode, J., S. Goetze, H. Heng, S. A. Krawetz and C. Benham. 2003. From DNA structure to gene expression: mediators of nuclear compartmentalization and dynamics. Chromosome Res. **11**: 435–45.

10 Bohjanen, P. R., Y. Liu and M. A. Garcia-Blanco. 1997. TAR RNA decoys inhibit tat-activated HIV-1 transcription after preinitiation complex formation. Nucleic Acids Res. **25**: 4481–6.

11 Boutillier, A. L., D. Monnier, D. Lorang, J. R. Lundblad, J. L. Roberts and J. P. Loeffler. 1995. Corticotropin-releasing hormone stimulates proopiomelanocortin transcription by cFos-dependent and - independent pathways: characterization of an AP1 site in exon 1. Mol. Endocrinol. **9**: 745–55.

12 Bucher, P. 1990. Weight matrix descriptions of four eukaryotic RNA polymerase II promoter elements derived from 502 unrelated promoter sequences. J. Mol. Biol. **212**: 563–78.

13 Bussemaker, H. J., H. Li and E. D. Siggia. 2001. Regulatory element detection using correlation with expression. Nat. Genet. **27**: 167–71.

14 Cam, H., E. Balciunaite, A. Blais, A. Spektor, R. C. Scarpulla, R. Young, Y. Kluger and B. D. Dynlacht. 2004. A common set of gene regulatory networks links metabolism and growth inhibition. Mol. Cells **16**: 399–411.

15 Cartharius, K., K. Frech, K. Grote, *et al.* 2005. MatInspector and beyond: promoter analysis based on transcription factor binding sites. Bioinformatics **21**: 2933–42.

16 Cavener, D. R. 1987. Comparison of the consensus sequence flanking translational start sites in *Drosophila* and vertebrates. Nucleic Acids Res. **15**: 1353–61.

17 Chen, Q. K., G. Z. Hertz and G. D. Stormo. 1995. MATRIX SEARCH 1.0: a computer program that scans DNA sequences for transcriptional elements using a database of weight matrices. Comput. Appl. Biosci. **11**: 563–6.

18 Chen, Q. K., G. Z. Hertz and G. D. Stormo. 1997. PromFD 1.0: a computer program that predicts eukaryotic Pol II promoters using strings and IMD matrices. Comput. Appl. Biosci. **13**: 29–35.

19 Chen, S., T. Itoh, K. Wu, D. Zhou and C. Yang. 2002. Transcriptional regulation of aromatase expression in human breast tissue. J. Steroid. Biochem. Mol. Biol. **83**: 93–9.

20 Chetouani, F., P. Monestie, P. Thebault, C. Gaspin and B. Michot. 1997. ESSA: an integrated and interactive computer tool for analyzing RNA secondary structure. Nucleic Acids Res. **25**: 3514–22.

21 Colgan, J. and J. L. Manley. 1995. Cooperation between core promoter elements influences transcriptional activity *in vivo*. Proc. Natl Acad. Sci. USA **92**: 1955–9.

22 Conaway, J. W. and R. C. Conaway. 1991. Initiation of eukaryotic messenger RNA synthesis. J Biol Chem **266**: 17721–4.

23 Cramer, P., C. G. Pesce, F. E. Baralle and A. R. Kornblihtt. 1997. Functional association between promoter structure and transcript alternative splicing. Proc. Natl Acad. Sci. USA **94**: 11456–60.

24 Davuluri, R. V., I. Grosse and M. Q. Zhang. 2001. Computational identification of promoters and first exons in the human genome. Nat. Genet. **29**: 412–7.

25 Doehr, S., A. Klingenhoff, H. Maier, M. Hrabe de Angelis, T. Werner and R. Schneider. 2005. Linking disease-associated genes to regulatory networks via promoter organization. Nucleic Acids Res. **33**: 864–72.

26 Down, T. A. and T. J. Hubbard. 2002. Computational detection and location of transcription start sites in mammalian genomic DNA. Genome Res. **12**: 458–61.

27 Dvir, A., J. W. Conaway and R. C. Conaway. 2001. Mechanism of transcription initiation and promoter escape by RNA polymerase II. Curr. Opin. Genet. Dev. **11**: 209–14.

28 FICKETT, J. W. 1996. Coordinate positioning of MEF2 and myogenin binding sites. Gene **172**: GC19–32.

29 FICKETT, J. W. AND A. G. HATZIGEORGIOU. 1997. Eukaryotic promoter recognition. Genome Res. **7**: 861–78.

30 FICKETT, J. W. AND W. W. WASSERMAN. 2000. Discovery and modeling of transcriptional regulatory regions. Curr. Opin. Biotechnol. **11**: 19–24.

31 FRECH, K., J. DANESCU-MAYER AND T. WERNER. 1997. A novel method to develop highly specific models for regulatory units detects a new LTR in GenBank which contains a functional promoter. J. Mol. Biol. **270**: 674–87.

32 FRECH, K., P. DIETZE AND T. WERNER. 1997. ConsInspector 3.0: new library and enhanced functionality. Comput. Appl. Biosci. **13**: 109–10.

33 FRECH, K., G. HERRMANN AND T. WERNER. 1993. Computer-assisted prediction, classification, and delimitation of protein binding sites in nucleic acids. Nucleic Acids Res. **21**: 1655–64.

34 FRECH, K., K. QUANDT AND T. WERNER. 1997. Finding protein-binding sites in DNA sequences: the next generation. Trends. Biochem. Sci. **22**: 103–4.

35 FRECH, K., K. QUANDT AND T. WERNER. 1998. Muscle actin genes: a first step towards computational classification of tissue specific promoters. In Silico Biol. **1**: 29–38.

36 FRECH, K., K. QUANDT AND T. WERNER. 1997. Software for the analysis of DNA sequence elements of transcription. Comput. Appl. Biosci. **13**: 89–97.

37 FRISCH, M., K. FRECH, A. KLINGENHOFF, K. CARTHARIUS, I. LIEBICH AND T. WERNER. 2002. *In silico* prediction of scaffold/matrix attachment regions in large genomic sequences. Genome Res. **12**: 349–54.

38 GALM, O. AND M. ESTELLER. 2004. Beyond genetics – the emerging role of epigenetic changes in hematopoietic malignancies. Int. J. Hematol. **80**: 120–7.

39 GAUTIER, L., M. MOLLER, L. FRIIS-HANSEN AND S. KNUDSEN. 2004. Alternative mapping of probes to genes for Affymetrix chips. BMC Bioinformatics **5**: 111.

40 GIANGRANDE, P. H., W. ZHU, R. E. REMPEL, N. LAAKSO AND J. R. NEVINS. 2004. Combinatorial gene control involving E2F and E Box family members. EMBO J. **23**: 1336–47.

41 GILINGER, G. AND J. C. ALWINE. 1993. Transcriptional activation by simian virus 40 large T antigen: requirements for simple promoter structures containing either TATA or initiator elements with variable upstream factor binding sites. J. Virol. **67**: 6682–8.

42 HENG, H. H., S. GOETZE, C. J. YE, et al. 2004. Chromatin loops are selectively anchored using scaffold/matrix-attachment regions. J. Cell Sci. **117**: 999–1008.

43 HOFACKER, I. L., B. PRIWITZER AND P. F. STADLER. 2004. Prediction of locally stable RNA secondary structures for genome-wide surveys. Bioinformatics **20**: 186–90.

44 HUTCHINSON, G. B. 1996. The prediction of vertebrate promoter regions using differential hexamer frequency analysis. Comput. Appl. Biosci. **12**: 391–8.

45 IMPEY, S., S. R. MCCORKLE, H. CHA-MOLSTAD, et al. 2004. Defining the CREB regulon: a genome-wide analysis of transcription factor regulatory regions. Cell **119**: 1041–54.

46 IOSHIKHES, I., A. BOLSHOY, K. DERENSHTEYN, M. BORODOVSKY AND E. N. TRIFONOV. 1996. Nucleosome DNA sequence pattern revealed by multiple alignment of experimentally mapped sequences. J. Mol. Biol. **262**: 129–39.

47 JAMORA, C., R. DASGUPTA, P. KOCIENIEWSKI AND E. FUCHS. 2003. Links between signal transduction, transcription and adhesion in epithelial bud development. Nature **422**: 317–22.

48 JIANG, J. G. AND R. ZARNEGAR. 1997. A novel transcriptional regulatory region within the core promoter of the hepatocyte growth factor gene is responsible for its inducibility by cytokines via the C/EBP family of transcription factors. Mol. Cell. Biol. **17**: 5758–70.

49 Johnson, J. M., J. Castle, P. Garrett-Engele, et al. 2003. Genome-wide survey of human alternative pre-mRNA splicing with exon junction microarrays. Science **302**: 2141–4.

50 Keich, U. and P. A. Pevzner. 2002. Finding motifs in the twilight zone. Bioinformatics **18**: 1374–81.

51 Kel-Margoulis, O. V., A. G. Romashchenko, N. A. Kolchanov, E. Wingender and A. E. Kel. 2000. COMPEL: a database on composite regulatory elements providing combinatorial transcriptional regulation. Nucleic Acids Res. **28**: 311–5.

52 Kel, A., O. Kel-Margoulis, V. Babenko and E. Wingender. 1999. Recognition of NFATp/AP-1 composite elements within genes induced upon the activation of immune cells. J. Mol. Biol. **288**: 353–76.

53 Kel, A. E., Y. V. Kondrakhin, A. Kolpakov Ph, O. V. Kel, A. G. Romashenko, E. Wingender, L. Milanesi and N. A. Kolchanov. 1995. Computer tool FUNSITE for analysis of eukaryotic regulatory genomic sequences. Proc. ISMB **3**: 197–205.

54 Kim, J., S. Klooster and D. J. Shapiro. 1995. Intrinsically bent DNA in a eukaryotic transcription factor recognition sequence potentiates transcription activation. J. Biol. Chem. **270**: 1282–8.

55 Kleinjan, D. A. and V. van Heyningen. 2005. Long-range control of gene expression: emerging mechanisms and disruption in disease. Am. J. Hum. Genet. **76**: 8–32.

56 Klingenhoff, A., K. Frech, K. Quandt and T. Werner. 1999. Functional promoter modules can be detected by formal models independent of overall nucleotide sequence similarity. Bioinformatics **15**: 180–6.

57 Kodadek, T. 1998. Mechanistic parallels between DNA replication, recombination and transcription. Trends Biochem. Sci. **23**: 79–83.

58 Lavorgna, G., A. Guffanti, G. Borsani, A. Ballabio and E. Boncinelli. 1999. TargetFinder: searching annotated sequence databases for target genes of transcription factors. Bioinformatics **15**: 172–3.

59 Lee, G. E., J. H. Kim and I. K. Chung. 1998. Topoisomerase II-mediated DNA cleavage on the cruciform structure formed within the 5′ upstream region of the human beta-globin gene. Mol. Cells **8**: 424–30.

60 Levy, S., L. Compagnoni, E. W. Myers and G. D. Stormo. 1998. Xlandscape: the graphical display of word frequencies in sequences. Bioinformatics **14**: 74–80.

61 Levy, S. and S. Hannenhalli. 2002. Identification of transcription factor binding sites in the human genome sequence. Mamm. Genome **13**: 510–4.

62 Li, H. and Y. Capetanaki. 1994. An E box in the desmin promoter cooperates with the E box and MEF-2 sites of a distal enhancer to direct muscle-specific transcription. EMBO J. **13**: 3580–9.

63 Lian, J. B., J. L. Stein, G. S. Stein, M. Montecino, A. J. van Wijnen, A. Javed and S. Gutierrez. 2001. Contributions of nuclear architecture and chromatin to vitamin D-dependent transcriptional control of the rat osteocalcin gene. Steroids **66**: 159–70.

64 Loots, G. G. and I. Ovcharenko. 2004. rVISTA 2.0: evolutionary analysis of transcription factor binding sites. Nucleic Acids Res. **32**: W217–21.

65 Mahony, S., D. Hendrix, A. Golden, T. J. Smith and D. S. Rokhsar. 2005. Transcription factor binding site identification using the self-organizing map. Bioinformatics.

66 Novina, C. D. and A. L. Roy. 1996. Core promoters and transcriptional control. Trends Genet. **12**: 351–5.

67 Ohler, U. and H. Niemann. 2001. Identification and analysis of eukaryotic promoters: recent computational approaches. Trends Genet. **17**: 56–60.

68 Panne, D., T. Maniatis and S. C. Harrison. 2004. Crystal structure of ATF-2/c-Jun and IRF-3 bound to the interferon-beta enhancer. EMBO J. **23**: 4384–93.

69 Pilpel, Y., P. Sudarsanam and G. M. Church. 2001. Identifying regulatory networks by combinatorial analysis of promoter elements. Nat. Genet. **29**: 153–9.

70 PRESTRIDGE, D. S. 1996. SIGNAL SCAN 4.0: additional databases and sequence formats. Comput. Appl. Biosci. **12**: 157–60.

71 PRITSKER, M., Y. C. LIU, M. A. BEER AND S. TAVAZOIE. 2004. Whole-genome discovery of transcription factor binding sites by network-level conservation. Genome Res. **14**: 99–108.

72 QUANDT, K., K. FRECH, H. KARAS, E. WINGENDER AND T. WERNER. 1995. MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data. Nucleic Acids Res. **23**: 4878–84.

73 RHEE, K. AND E. A. THOMPSON. 1996. Glucocorticoid regulation of a transcription factor that binds an initiator-like element in the murine thymidine kinase (Tk-1) promoter. Mol. Endocrinol. **10**: 1536–48.

74 ROTHENBERG, E. V. AND S. B. WARD. 1996. A dynamic assembly of diverse transcription factors integrates activation and cell-type information for interleukin 2 gene regulation. Proc. Natl Acad. Sci. USA **93**: 9358–65.

75 SAP, J., A. MUNOZ, J. SCHMITT, H. STUNNENBERG AND B. VENNSTROM. 1989. Repression of transcription mediated at a thyroid hormone response element by the v-*erb-A* oncogene product. Nature **340**: 242–4.

76 SCHERF, M., A. KLINGENHOFF, K. FRECH, et al. 2001. First pass annotation of promoters on human chromosome 22. Genome Res. **11**: 333–40.

77 SCHERF, M., A. KLINGENHOFF AND T. WERNER. 2000. Highly specific localization of promoter regions in large genomic sequences by PromoterInspector: a novel context analysis approach. J. Mol. Biol. **297**: 599–606.

78 SCHMID, C. D., V. PRAZ, M. DELORENZI, R. PERIER AND P. BUCHER. 2004. The Eukaryotic Promoter Database EPD: the impact of *in silico* primer extension. Nucleic Acids Res. **32**: D82–5.

79 SINGH, G. B., J. A. KRAMER AND S. A. KRAWETZ. 1997. Mathematical model to predict regions of chromatin attachment to the nuclear matrix. Nucleic Acids Res. **25**: 1419–25.

80 SINHA, S., M. BLANCHETTE AND M. TOMPA. 2004. PhyME: a probabilistic algorithm for finding motifs in sets of orthologous sequences. BMC Bioinformatics **5**: 170.

81 SLOAN, L. S. AND A. SCHEPARTZ. 1998. Sequence determinants of the intrinsic bend in the cyclic AMP response element. Biochemistry **37**: 7113–8.

82 SOLOVYEV, V. AND A. SALAMOV. 1997. The Gene-Finder computer tools for analysis of human and model organisms genome sequences. Proc. ISMB **5**: 294–302.

83 STADEN, R. 1984. Computer methods to locate signals in nucleic acid sequences. Nucleic Acids Res. **12**: 505–19.

84 STEIN, G. S., A. J. VAN WIJNEN, J. STEIN, J. B. LIAN AND M. MONTECINO. 1995. Contributions of nuclear architecture to transcriptional control. Int. Rev. Cytol. **162A**: 251–78.

85 STORMO, G. D. 2000. DNA binding sites: representation and discovery. Bioinformatics **16**: 16–23.

86 STORMO, G. D. AND G. W. HARTZELL, 3RD. 1989. Identifying protein-binding sites from unaligned DNA fragments. Proc. Natl Acad. Sci. USA **86**: 1183–7.

87 SUZUKI, Y., R. YAMASHITA, S. SUGANO AND K. NAKAI. 2004. DBTSS, DataBase of Transcriptional Start Sites: progress report 2004. Nucleic Acids Res. **32**: D78–81.

88 TOMPA, M., N. LI, T. L. BAILEY, et al. 2005. Assessing computational tools for the discovery of transcription factor binding sites. Nat. Biotechnol. **23**: 137–44.

89 TRONCHE, F., F. RINGEISEN, M. BLUMENFELD, M. YANIV AND M. PONTOGLIO. 1997. Analysis of the distribution of binding sites for a tissue-specific transcription factor in the vertebrate genome. J. Mol. Biol. **266**: 231–45.

90 USHEVA, A. AND T. SHENK. 1996. YY1 transcriptional initiator: protein interactions and association with a DNA site containing unpaired strands. Proc. Natl Acad. Sci. USA **93**: 13571–6.

91 VAN DRUNEN, C. M., R. G. SEWALT, R. W. OOSTERLING, P. J. WEISBEEK, S. C.

SMEEKENS AND R. VAN DRIEL. 1999. A bipartite sequence element associated with matrix/scaffold attachment regions. Nucleic Acids Res. **27**: 2924–30.

92 WANG, W., T. CHI, Y. XUE, S. ZHOU, A. KUO AND G. R. CRABTREE. 1998. Architectural DNA binding by a high-mobility-group/kinesin-like subunit in mammalian SWI/SNF-related complexes. Proc. Natl Acad. Sci. USA **95**: 492–8.

93 WASSERMAN, W. W. AND J. W. FICKETT. 1998. Identification of regulatory regions which confer muscle-specific gene expression. J. Mol. Biol. **278**: 167–81.

94 WASSERMAN, W. W., M. PALUMBO, W. THOMPSON, J. W. FICKETT AND C. E. LAWRENCE. 2000. Human–mouse genome comparisons to locate regulatory sites. Nat. Genet. **26**: 225–8.

95 WERNER, T. 2001. Cluster analysis and promoter modelling as bioinformatics tools for the identification of target genes from expression array data. Pharmacogenomics **2**: 25–36.

96 WERNER, T. 1999. Models for prediction and recognition of eukaryotic promoters. Mamm. Genome **10**: 168–75.

97 YAMAUCHI, M., Y. OGATA, R. H. KIM, J. J. LI, L. P. FREEDMAN AND J. SODEK. 1996. AP-1 regulation of the rat bone sialoprotein gene transcription is mediated through a TPA response element within a glucocorticoid response unit in the gene promoter. Matrix Biol. **15**: 119–30.

98 YELIN, R., D. DAHARY, R. SOREK, et al. 2003. Widespread occurrence of antisense transcription in the human genome. Nat. Biotechnol. **21**: 379–86.

99 ZAVOLAN, M., S. KONDO, C. SCHONBACH, J. ADACHI, D. A. HUME, Y. HAYASHIZAKI AND T. GAASTERLAND. 2003. Impact of alternative initiation, splicing, and termination on the diversity of the mRNA transcripts encoded by the mouse transcriptome. Genome Res. **13**: 1290–300.

100 ZAWEL, L. AND D. REINBERG. 1995. Common themes in assembly and function of eukaryotic transcription complexes. Annu. Rev. Biochem. **64**: 533–61.

7 Finding Repeats in Genome Sequences

Brian J. Haas and Steven L. Salzberg

1 Introduction

An essential component of genome sequence analysis is the identification of repetitive sequences (repeats). A repeat is a substring that occurs multiple times within a sequence or collection of sequences. Repeats are commonly found in the genomes of both prokaryotes and eukaryotes, although generally to a lesser extent in the compact genomes of prokaryotes. In some cases, the number of repeats and their contribution to overall genome size and content is staggering, e.g. ;ore than half of the human genome is composed of repetitive sequences [34]. In addition, the large genomes of higher plants including maize and wheat are composed mostly of repetitive sequences [3, 7, 19]. In some bacteria, repetitive plasmid sequences are so similar to one another that it is extremely difficult even to determine how many plasmids are present, as in the case of the Lyme disease spirochete [20].

Although repeated sequences represent a diverse group of features, they tend to fall into one of two broad categories: tandem repeats or dispersed repeats. Tandem repeats are those that are found directly adjacent to one another, contiguously arrayed. These are often termed "satellite" DNA. Simple sequence repeats (SSRs or microsatellites) are tandem repeats where the repeat unit is very short, typically 1–6 nucleotides. SSRs tend to have a uniform distribution within genomes, and are sometimes found within protein coding and untranslated regions of genes. Trinucleotide repeats within genes are of special interest since they have been linked to several human genetic disorders, including Fragile-X mental retardation, Huntington's disease and myotonic dystrophy [37, 50, 51]. The term "satellite repeat" typically refers to repeat units greater than 100 bp, which are found as contiguous stretches that can span up to tens of thousands or even millions of base pairs of chromosomal DNA. Such satellite repeats include the 170- to 180-bp repeat units found at centromeres of higher eukaryotes [12, 23] and the long contiguous rDNA cassettes that comprise nucleolus organizer regions [47]. The term

Bioinformatics - From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

minisatellites is used to refer to tandem repeats with unit lengths intermediate to SSRs and satellites.

Dispersed repeats predominantly consist of transposable elements – mobile sequences that can cut and paste or copy themselves to other locations in a genome (for details, see Refs. [11, 13]). Complete autonomous transposons encode one or more proteins that are required for their mobility, and can exceed 10 kb in length. Nonautonomous transposable elements also parasitize genomes; these numerous elements lack the machinery for their own transposition and rely on the proteins encoded by other complete elements to mediate their transposition. Transposons tend to be most abundant in regions of heterochromatin, typically in pericentromeric regions mostly devoid of expressed genes. These elements are sometimes found within introns of genes or interrupting an exon, in which case the gene is likely rendered nonfunctional. Dispersed repeats account for a majority of the repeat content in large eukaryotic genomes. Little is known about the purpose of transposons; they are often regarded as "selfish" elements that provide no benefit to the host organism.

Molecular events resulting in gene duplication, including unequal crossing over during meiotic recombination, recombinational repair or, at the extreme, whole genome duplications, also generate repeated sequences. On the smallest level, slippage of the DNA replication machinery can result in short repeats gaining additional copies. Depending on the event responsible for the genomic rearrangement and the resulting configuration of the genetic material, the duplicated segments may appear in tandem or at remote locations typical of dispersed repeat families. Repeat regions are involved in multiple human diseases, the most well-known being Down's syndrome, which involves an extra copy of chromosome 21. Both Huntington's disease and Fragile X syndrome result from an expansion of trinucleotide repeats. At another level, repeats are used for the new science of microbial forensics; as these regions are among the most highly variable in many species, they provide unique DNA-based signatures that distinguish bacteria from one another, including very closely related strains of organisms such as the anthrax bacterium, *Bacillus anthracis* [49].

Rigorous studies of genome sequence repeats involve identifying similar sequence pairs, grouping the related elements to examine their number and distribution within the genome, differentiating repeats of known function from those of unknown function (and genes from nongenes), and unraveling the details of the length, number of copies and orientation of repeat elements. Each step of the analysis is complicated by the nature of the underlying repetitive sequences, including the degree of divergence between related elements, the background of genomic architectural rearrangements which disrupt or conceal the original repeats and the resulting mosaic nature of repeat elements such that related repeats may share only a subsequence in common. In particular, the boundaries of repeats are notoriously difficult to resolve, complicated by issues described above coupled with the difficulty in obtaining pairwise alignments which terminate precisely at repeat boundaries. An often-cited statement made by Bao and Eddy [4] nicely summarizes the state of automated repeat finding: “the problem of automated repeat sequence family classification is inherently messy and ill-defined and does not appear to be amenable to clean algorithmic attack”. This remains true today, although new algorithms, tools and ideas regarding repeat analysis continue to shed light on the problem.

Repetitive sequences impose formidable challenges to sequence analysis in the postgenomic era. They create havoc for genome assembly; regions rich in repeats are difficult if not impossible to assemble correctly using currently available tools and algorithms, and often lead to misassembly of regions flanked by repeats or excessive fragmentation of what would otherwise be a more cohesive genome sequence (see also Chapter 2). Subsequent to sequence assembly, genome annotation is also confounded by repeats. In particular, the transposon sequences found between genes and within introns can be easily mistaken for exons of protein-coding genes by gene-finding programs. This “junk” DNA requires prior recognition and exclusion to facilitate more accurate identification of the coveted host genes localized to the remaining sequence (see also Chapter 5).

This chapter focuses on the algorithms and tools commonly used for identifying repeats in genome sequences. The impact of repeats on genome assembly and methods used by assemblers to circumvent associated problems are described. Additional topics include methods for clustering elements to organize repeat families, resolving repeat boundaries, efforts to untangle the mosaic nature of related repeats and the annotation of repeat sequences.

2 Algorithms and Tools for Mining Repeats

Sequence alignment is at the very core of repeat identification. In contrast to aligning sequences from different genomes to identify regions of homology, sequence alignment is applied to a single genome, as a single sequence or collection of sequences, to identify significant intra- and inter-sequence similarities. The more general application of repeat analysis is summarized as first finding all pairwise alignments, then clustering related elements. Finding all pairwise alignments is relegated to standard sequence alignment tools and algorithms, a topic described in Chapter 3 and so minimally covered here. Clustering of repeat elements into repeat families is a major challenge, and recent efforts to deconvolute pairwise alignments into more meaningful repeat sets are described. Finding tandem repeats is a related, but distinct challenge; a vast amount of literature exists on this topic, describing algorithms and tools which are specially designed for this aspect of repeat structure. Because of this, tandem repeats are the focus of a separate section in this chapter.

2.1 Finding Intra- and Inter-sequence Repeats as Pairwise Alignments

Pairwise sequence alignment algorithms are well suited to the problem of repeat identification; due to the enormous complexity of the problem in large genomes, heuristics are required to improve efficiency. The Smith–Waterman alignment algorithm [54] finds the single best-scoring local alignment between two sequences. If the sequences being compared are two distinct entries from the collection of sequences corresponding to the single genome under study, this best local alignment would suffice as a repeat. This approach cannot be used, however, when the genome is a single contiguous sequence, as is often the case for complete bacterial, archaeal and viral genomes. Comparing a sequence to itself to find the best local alignment would yield only the obvious perfect alignment along the diagonal corresponding to the alignment of the sequence matching itself from beginning to end. A modification to the Smith–Waterman algorithm, as described by Waterman and Eggert [58], affords the identification of all nonintersecting high scoring alignments, rather than just the single best local alignment between two sequences (see also Chapter 3). This modification unravels the internal repetitive structure of a sequence when aligned to itself. Huang and Miller [24] describe the sim algorithm, which yields all high scoring nonintersecting alignments using linear space, and the lalign utility of Bill Pearson's fasta2 toolsuite (http://ftp.virginia.edu/pub/fasta/) is a popular tool that implements this algorithm. The accompanying plalign utility generates an illustration of the repetitive structure as a postscript file.

Although these algorithms are well suited to repeat finding, they are simply not fast enough to tackle large genomes and so we turn to heuristics. The "seed and extend" heuristic is perhaps the most common strategy to quickly ascertain significant pairwise alignments between sequences. Early uses of this strategy include the FASTA algorithm [43], followed by the hugely popular BLAST algorithm [1, 2], among other database search and alignment tools including MUMmer [14, 15, 32], PatternHunter [35, 36] and BLAT [29], and a less well known but similarly useful tool for studying repeats called ICAass [41], the repeat mining utility of the Miropeats software [42]. Here, matches to exact words of predefined length provide the seeds for alignments, which are extended in both directions to extract the maximal scoring alignment containing the seed (see also Chapter 3). A major limitation of these methods is the requirement of a predefined seed length. A seed length that is too short requires numerous extensions, few of which lead to significant alignments. A seed length that is too long involves fewer extensions, but many significant alignments may lack such a seed and are missed. The programs MUMmer and Reputer [31, 33] take a more sophisticated approach, employing a suffix tree data structure to find exact word matches. The suffix tree is not limited to finding seeds of a constant length; all exact matches are found regardless of length and this is done very fast, in linear time and space.

The focus of repeat-finding software can vary; the focus may be to find all matching substrings within a sequence or among a collection of sequences, or the focus may be the postprocessing of pairwise alignments to cluster related elements into families, resolve repeat boundaries or to illustrate the mosaic nature of repeated sequences. Progress in repeat analysis, as in other areas, builds upon previous contributions to the field. As such, we present an overview of each contribution in roughly chronological order.

2.2 Miropeats (alias Printrepeats)

Miropeats [42] is perhaps one of the earliest and most popular repeat-finding analysis tools to find widespread use in genome sequence analysis (see, e.g. Refs [52]). The repeat-finding engine of Miropeats is the program ICAass of the ICAtools suite [41]. ICAass finds maximal gap-less aligned segment pairs (MSPs) within a single sequence and/or among a collection of sequences using a "seed and extend" strategy similar to that used in BLASTN [1b]. All overlapping 8-mers are loaded into a hash table, using two-bits per base encoding and so allowing 4 bases per byte. All sequences to be examined are indexed and the 8-mers are then used to seed potentially longer alignments. Those ungapped alignments meeting the minimum score threshold are reported. While the ICAass program includes additional components, only the MSP identification steps are utilized by Miropeats. Miropeats is a Unix C-shell script that calls ICAass to identify MSPs as repeats, and then writes a postscript file which illustrates the positions and associations among the repeats. Arcs are drawn between the matching end-points of each repeat pair and arcs are drawn in such a way to help ascertain their relative orientation and overlap. Although ICAass is used, in theory, any program capable of generating meaningful pairwise alignments could be employed, including BLAST (i.e. WU-BLAST with the -span option selected), BLAT or PatternHunter, although Miropeats would require some minor customization to accept this. The strength of Miropeats as a repeat analysis tool lies in its illustration capabilities, particularly with respect to those repeats found in close proximity along the nucleotide sequence, e.g. the structures of transposable elements typically include some form of terminal repeat, either direct or inverted, at the elements' boundaries. The illustration of repeats within transposon-rich regions helps to elucidate their terminal repeat structures as well as their relative abundance along the genomic contig; an example is provided in Figure 1, where the long terminal repeats (LTRs) of a gypsy-family retrotransposon are nicely illustrated by the Miropeats software). (A "contig" is a contiguous stretch of DNA without gaps. Whole-genome shotgun (WGS) sequencing projects generally produce nearly complete genomes that consist of a set of contigs separated by gaps.) As with other repeat-finding applications, the use of Miropeats extends beyond repeat analysis and includes additional areas of sequence comparison, such as to position a small set of bacterial artificial chromosomes (BACs) in a section of a genome BAC tiling path by defining the overlaps among their ends (see also Chapter 2). Due to the relatively slow ICAass repeat-finding step and because of the static illustration of the repeat structures provided by Miropeats, the software is limited in practice to analyzing sequences whose length is no more than a few hundred thousand base pairs, although application to longer sequences is not restricted.

2.3 REPuter

Of the methods available for finding repeated strings in genomic sequences, those based on suffix trees are most efficient and practical for large-scale genome analyses. A suffix tree is a data structure specifically designed to capture a text string and all of its substrings, which makes it well suited for capturing DNA and protein sequences. The tree itself is set of nodes and edges, where each edge is labeled with a string. A suffix of a string *S* (which might be an entire genome, for example) is simply a substring that starts within *S* and extends to the end of *S*. A suffix tree represents all suffixes of *S* implicitly; each suffix is a path from the root node to a leaf node of *S*. Internal nodes of *S* represent other (nonsuffix) substrings; in fact, a suffix tree contains all substrings of *S*. Suffix trees represented a major advance over previous sequence analysis techniques because of two key properties: (i) the size of the tree is a linear function of the size of the sequence, and (ii) the tree can be constructed and searched in linear time. This contrasts with alternative sequence alignment methods, which are quadratic in time, space or both. Details regarding construction and search algorithms for suffix trees are described in Ref. [22].

Stefan Kurtz's REPuter software [31,33], the first production quality repeat-finding software to employ suffix trees, can mine complete eukaryotic genomes (megabase pairs) for all maximal repeats in a matter of seconds on a personal computer. REPuter has existed in two versions. The earlier version, first described in 1999, was limited to finding identical maximal repeats. An enhanced tool suite was released under the package name REPuter in 2001 with the search engine named REPfind, capable of extending the problem of

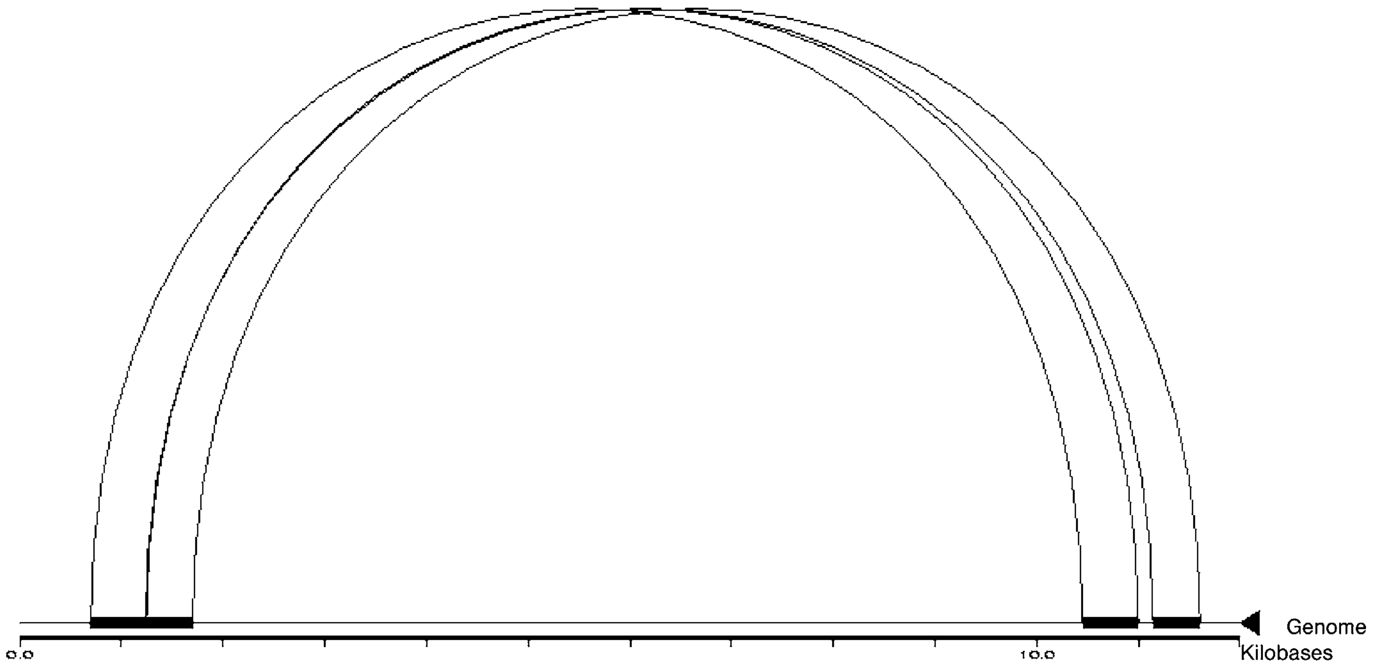


Figure 1 LTRs of gypsy-like retrotransposon family (Athila) element of *Arabidopsis* (locus At1g40077) found and illustrated using Miropeats. Arcs are drawn connecting pairwise matches found in an approximately 12-kb stretch of genome sequence, corresponding here to the LTRs of an *Arabidopsis* gypsy-family retrotransposon.

repeat finding from identical repeats only, to approximate repeats, allowing for mismatches and indels. Each version introduced key concepts and contributions to repeat analysis, so each is described here in order of their availability. To distinguish between the two versions, the earlier version is referred to as REPuter and the later version as REPfind.

As stated earlier, REPuter is limited to finding exactly identical maximal repeats; the repeats are maximal in that extending the alignment between two paired sequence regions would introduce a mismatch and violate the requirement of identical sequence pairs. REPuter can find the following four classifications of repeats, each exemplified using the 4-mer "gcta" and the forward sequence orientation (top strand only):

Forward:	5′-gcta-3′ with 5′-gcta-3′
Palindromic (reverse complemented or inverted):	5′-gcta-3′ with 5′-tagc-3′
Complemented:	5′-gcta-3′ with 5′-cgat-3′
Reversed:	5′-gcta-3′ with 5′-atcg-3′

Although cataloguing all identical sequence substrings is a useful component of repeat analysis, few repeats, unless very recently duplicated, will be free of mismatches or indels. As a result, these repeated identical substrings are often parts of larger repeat units that are nonidentical, although detectibly similar in sequence. REPfind takes this into account and is able to find degenerate repeats, allowing for mismatches and insertions/deletions (indels) as part of the larger repeats.

REPfind exhaustively finds all degenerate repeats in a genome sequence given a user specified minimum length and maximum number of errors. Errors are measured by one of two methods: hamming distance or edit distance. Hamming distance corresponds to the number of mismatches in a gap-free sequence alignment. Edit distance includes the number of differences in an alignment possibly containing indels. The identification of approximate repeats relies on the basis that every degenerate repeat contains a substring of identical sequence.

To find approximate repeats, REPfind locates all exact word matches followed by an extension process to determine if the word match is part of a longer degenerate repeat. Approximate repeats of two types are found: maximal mismatch repeats using the MMR algorithm and maximal difference repeats using the MDR algorithm, as described below. Both types of repeats rely on the existence of an exact word match; the exact word matches are found as described earlier by the original REPuter software.

The MMR algorithm finds a gap-less maximal mismatch repeat by looking for the longest alignment that contains the seed and has no more than k mismatches (the degenerate repeat in this context called a maximal k-mismatch repeat; here k is a specified parameter. This is done by identifying the first $k + 1$

mismatched nucleotides to the left of the seed [ordered from left to right (l_1, $l_2, \ldots, l_{(k+1)}$)], followed by identifying the first $k + 1$ mismatches to the right of the seed (ordered r_1, r_2, …, $r_{(k+1)}$), with the mismatches l_1 and $r_{(k+1)}$ bounding a sequence region of k mismatches from the left or right of the seed boundary, respectively. For all values of i, from 1 to $k + 1$, the substring with coordinates $l_i + 1$ to $r_i - 1$ contains exactly k mismatches. The k-mismatch substring with the greatest length is reported.

The MDR algorithm extends seeds taking into account insertions and deletions. The idea is similar to the MMR algorithm in that k-differences are explored to each side of the seed, and the combination of coordinates within this range which maximize repeat length and satisfy the k-mismatch criteria are chosen. The primary difference is that, instead of searching for nucleotide differences along a single dimension as with MMR, a search is performed in two dimensions allowing for insertions and deletions. A dynamic programming matrix banded at $\pm \pm (k + 1)$, extending from both ends of the seed, is used to find all alignment termini yielding each of 1 to k maximum number of mismatches. Each pair of alignment termini are examined and the pair of left and right termini providing the longest repeat length and a maximum of k-differences is reported.

It is often the case that a single maximal k-difference repeat will contain multiple seeds. To avoid outputting distinct maximal k-difference repeats which contain seeds of neighboring k-difference repeats, the alignment extensions to the left of a target seed are restricted to the right of any previously occurring seed. This guarantees that each maximal k-difference repeat will derive from the extension of its left-most containing seed.

By default, REPfind reports only exact matches, as done by the earlier REPuter program. Options are available to pursue either k-mismatch repeats using the MMR algorithm or k-difference repeats using the MDR algorithm. Unless there is a keen interest in obtaining gap-less repeats only, it is sensible to mine maximum difference repeats exclusively using the MDR algorithm, given that it will report maximum matches with or without gaps, whichever provides the maximal k-difference repeat. Rather than setting the k-value directly, the user can specify the parameter values of minimum repeat length and a maximum error rate, from which the value of k is computed internally.

An improvement over the earlier REPuter software is the inclusion of statistical significance for each of the repeats found in the form of an E-value (see Chapter 3 for an explanation of the concept of an E-value). In the case that multiple solutions exist for the maximal k-mismatch or k-difference repeat, the single repeat yielding the sequence with the smallest E-value is reported. By selecting the option "-allmax", each solution is reported in the case of ties among candidates meeting the maximum length k-difference criteria.

REPuter is available to researchers in several forms: a set of command-line driven utilities for local installations, a comfortable web interface for more interactive and targeted analyses and most recently as a web service enabling distributed computing environments with repeat analysis capabilities.

We should note that the REPuter package, although still widely used and available to researchers, has more recently been subsumed by the Vmatch large-scale sequence analysis software (Kurtz, unpublished; http://www.vmatch.de). Improvements include the use of suffix arrays in place of suffix trees, which reduces memory requirements and processing time. Also, the alphabet of sequences to be aligned is no longer restricted to nucleotide characters, allowing one to examine protein sequences as well.

2.4 RepeatFinder

Given fast and efficient methods to detect pairwise similarities within or among sequences, some repeat analysis software is devoted to the postprocessing of pairwise alignment data to collect and organize the repetitive sequences identified. An early example of this is the Repeat Pattern Toolkit (Agarwal and States [1a]) applied to the clustering of WU-BLAST ungapped alignments derived from 3.6 Mbp of the *Caenorhabditis elegans* genome, placing the alignments into a graph, and finding the minimum spanning tree for connected components to represent the relationships between repeats. A more modern approach involves the postprocessing of repeats found using suffix trees.

Natalia Volfovsky's RepeatFinder [56] uses a catalog of exactly repeated strings to further refine the definitions of individual repeated elements followed by the construction of repeat classes. In contrast to our canonical definition of a repeat as a pair of sequences which share similarity from beginning to end, RepeatFinder describes merged repeats where a merged repeat is found elsewhere in the genome at least once, and may be found in partial copies. The exactly repeated strings are found using the original REPuter software; a newer version of RepeatFinder uses REPfind. These exact matches compose the initial repeat set and these are redefined as repeat elements using a merging procedure. Since repeated sequences are expected to contain mismatches and indels, few complete repeats will be reported as exact matches. The merging procedure serves to consolidate regions defined as repeats that are found in close proximity or overlapping along each genomic sequence. By doing so, indels and mismatches fragmenting single repeats into disparate word matches are merged into larger degenerate repeats, and the dispersive and the fragmented nature of repeat regions is accounted for (i.e. portions of a larger repeat may be found as separate fragments elsewhere in the genome). The merging procedure to redefine repeat regions is restricted to either merging overlapping repeats, or merging neighboring repeats, with minimum overlap or gap size as user-specified parameters, respectively.

Each merged repeat retains a list of all the originally identified repeats (considered subrepeats) contained by it. Clustering of related merged repeats is done by grouping those merged repeats containing subrepeats in common into the same class. In order to further collapse clusters of similar elements, an "all-vs-all" BLASTN search is performed and separate clusters containing elements with sequence similarity (below a specified *E*-value threshold) are grouped into a single cluster.

Although the software is useful for rapidly extracting repetitive sequences from the genome and grouping related elements, the boundaries of the repeats remain ill-defined and all members of each cluster are not guaranteed to be similar to each other given the transitive relationships established via the clustering algorithm employed. More sophisticated clustering methods employed by RECON [4] address these issues more satisfactorily.

2.5 RECON

Bao and Eddy's development of RECON [4] for repeat analysis was viewed as a pioneering effort, as it represented the first tool to attempt to delineate boundaries of repeat elements in a biologically meaningful way. The algorithm of RECON is broken down into the following major tasks: obtaining pairwise alignments among the input sequences, defining elements based on the pairwise alignments and, finally, grouping elements into families. In contrast to RepeatFinder which obtains the pairwise alignment data using REPuter, RECON uses BLASTN of the WU-BLAST package [21]. The process of defining repeat elements based on pairwise alignment data is illustrated in Figure 2.

Repeat elements are initially defined by collapsing the overlapping pairwise alignments along the genomic sequence (Step II in Figure 2). Multiple alignment information is used to infer the boundaries of the element, and also to recognize and partition those elements found to be composed of multiple distinct repeat units. Given the set of overlapping alignments that initially define a repeat element in the genomic sequence, a preponderance of alignment ends found clustered to a short region of genomic sequence signifies a boundary of an element. Some candidate boundaries may be misleading because they derive from related but distinct repeat elements, those which share subrepeats in common, but are otherwise different. Misleading alignments between pairs of elements are identified by their proportionally large amount of unaligned sequence when compared to the entire element lengths (not shown). These are then discarded from subsequent element boundary refinement methods.

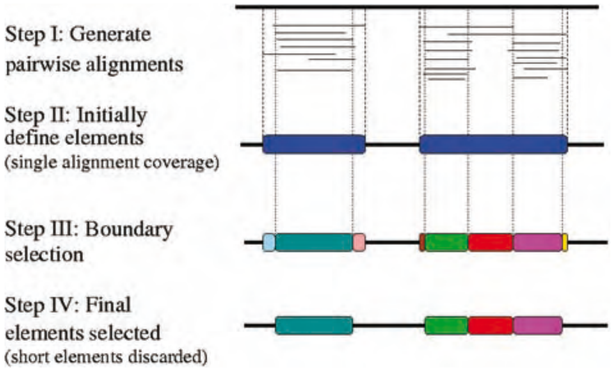

Step I: Generate pairwise alignments
Step II: Initially define elements (single alignment coverage)
Step III: Boundary selection
Step IV: Final elements selected (short elements discarded)

Figure 2 RECON's algorithm for defining repeat elements. In the first stage, WU-BLAST is used to generate pairwise alignments. These pairwise alignments are collapsed along the genomic sequence to define regions of alignment coverage. Clusters of alignment boundaries within short windows are used to redefine element boundaries and initial elements are repartitioned at these boundaries into separate elements. Short elements likely resulting from falsely extended alignments (from the first step) are removed to yield the refined final element set.

After eliminating the misleading alignments, the remainders are examined for the purpose of boundary refinement. Aggregations of alignment endpoints are identified by sliding a short window of predefined length (default 30 nucleotides) along the repeat element, clustering all neighboring alignment ends found separated by no more than the window size. The ratio of alignments with clustered ends to the total number of alignments spanning the corresponding region is used as an indicator of the significance of an aggregation point. A ratio above a specified threshold (default of 2.0) infers a boundary condition and the boundary is defined as the mean coordinate value for the clustered ends. Upon finding a significant aggregation point, the original element is considered composite. The composite element and its underlying supporting alignments are split at the boundary, and the split alignments are reassigned to their corresponding split element (Step III of Figure 2). Elements without significant aggregation points remain as originally defined. Split elements or the split supporting alignments found shorter than a minimum length cutoff are presumed artifacts due to the short random extensions that occur in pairwise alignments and these are discarded (Step IV of Figure 2). The remaining elements provide the set of repeats with defined boundaries.

Following the identification of the individual repeat elements as described above, the elements are classified into families. Special effort is taken to group related but distinct families separately. First, candidates for family membership are chosen by examining alignments between element pairs. For the purpose of clustering the elements, a graph is constructed in which elements are represented by nodes and relationships between nodes represented by edges. Edges are classified into two types: primary edges are used to link elements of the same family, those elements found to align to more than a specified threshold of length coverage of either element (default 90%); secondary edges link elements of different but related families that contain significant alignments but below the threshold of alignment coverage required for family membership. Before clustering family members based on the primary edges, all edges require reevaluation because some edges may have been falsely classified as primary edges. Partial elements are easily misclassified with primary edges during edge assignment since they pass the alignment coverage test with complete elements to which they are compared. It is by virtue of the secondary edges that false primary edges are identified and remedied. False primary edges are found via triangles of inequality: for example, elements A and B are deemed from the same family (primary edges), and elements A and C are deemed from the same family (primary edges), but elements B and C are deemed from separate families (secondary edge). In this case, element A is presumed partial given that it aligns with high coverage separately to the two elements B and C, which themselves lack significant coverage of alignments between them. To prevent element A from grouping the two related but distinct families together, all but the single primary edge extending from A, corresponding to its most similar element, are removed. Following the conversion of false primary edges to secondary edges, all the secondary edges are removed and families are generated by transitive closure of the remaining primary edges.

The algorithm of RECON addresses the problem of delineating the boundaries of individual repeat elements as they are found in the genomic sequence, but it does not describe the mosaic nature of the related repeat elements nor the consensus boundaries and length of a prototypical element among a family of elements. A rough consensus sequence for large RECON-defined repeat families can be derived from alignments of the longest repeat elements within each large family. This is a useful approximation and works well for some repeat families, but is not rigorous enough to yield a consensus for each repeat family in a biologically meaningful way.

2.6 PILER

As discussed in the Introduction, repeats are diverse features, with wide variety in size, location and biological function. Major classes of repeats, including dispersed or tandem repeats, yield specific patterns in the context of whole-genome self-alignments. Bob Edgar's PILER [16] includes a suite of tools each of which focuses on specific patterns evident in sets of alignments to reliably identify elements of the corresponding repeat class. Examples of patterns

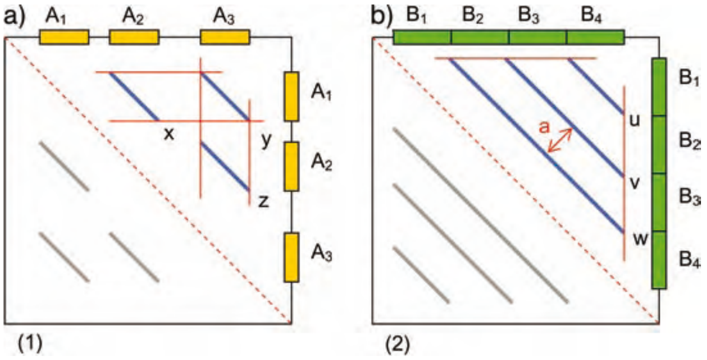


Figure 3 Patterns of sequence alignments targeted by PILER. Dot plots are shown for a comparison of a genome sequence against itself, with the dotted line as an indicator of the main diagonal of the plot. (a) Patterns of alignments generated by alignments of members of a family of dispersed repeats A_1, A_2 and A_3. (b) The "pyramid" pattern generated by alignments from a stretch of tandem repeats B_1 through B_4, indicating a repeat length of a. Figures were derived from Ref. [16], and reproduced here with permission from author Bob Edgar and Oxford University Press.

sought by PILER are illustrated in Figure 3. The individual tools of PILER and corresponding repeat classes are: PILER-DF for detecting individual intact elements of a dispersed repeat family, PILER-PS to find pseudo-satellites, PILER-TA to find tandem arrays, and PILER-TR to find repeat elements that have terminal repeats (a common characteristic of intact transposable elements).

Similarly to RepeatFinder, RECON, and other repeat finding and clustering tools, a set of intra-genome alignments is required. Rather than rely on REPUTER or BLAST to generate alignments, PILER includes an efficient alignment program called PALS (Pairwise Alignment of Long Sequences), which is specially designed with optimizations for detecting repetitive sequences; optimizations targeted towards searching a sequence against itself, limiting searches to banded regions and unrestricted reporting of numerous colocalized alignments, among others. After generating all alignments, overlapping hits along the genome sequence are linked together into a "pile" of contiguously overlapping alignments. These piles of hits are further subjected to specific analyses provided by the PILER-* utilites.

PILER-DF is designed to detect "Dispersed Families" of repeats with characteristics of transposable elements. The specific signature of a dispersed family as revealed by pairwise alignments is illustrated in Figure 3(a). The isolated elements are found as globally alignable regions, all with alignments of similar lengths. The goal of PILER-DF is to find aligned pairs that have similar characteristics. This is done by analyzing aligned sequence pairs such that each aligned region is found in a different pile (dispersed). Given a pairwise alignment between X and Y, such that X and Y are in different piles, an edge is drawn between X and Y if each aligned region spans most of its corresponding pile [(length of X)/(length of pile containing X), (length of Y)/(length of pile containing Y)]. After examining all pairwise alignments in this matter, all connected components are found in the resulting graph. The connected components are interpreted as dispersed families of complete repetitive elements.

PILER-PS searches for "Pseudo-Satellites; – repeats with features of satellite sequences in that they are found clustered locally in the genome. The algorithm here is identical to PILER-DF with the exception that the pairwise alignments result from a banded search, requiring that the alignments be in close proximity to one another.

PILER-TA finds "Tandem Arrays". The repeat-finding tools described so far are mostly limited or specially tuned to find dispersed repeats. PILER-TA is an exception in that it purposely mines these features from the genome. Please note that the general topic of finding tandem repeats is the major focus of the next section of this chapter and so it will be mentioned only briefly here as it relates to PILER. Sequences arrayed in tandem leave a specific signature in the pairwise alignments termed "pyramids" (see Figure 3b). The first observation is that pairwise alignments in a given pyramid are restricted to the same pile since all of them overlap. A banded search is used to find pairs of alignments within a pile that have the following characteristics: the shorter alignment pair is at least half the length of the longer alignment pair, and the distances between the alignments' respective start and end coordinates are each within a predefined percentage of the shorter alignment length. All pairs of alignments meeting such criteria are connected by an edge and, at completion, all connected components are gathered. Each connected component is interpreted as a tandem array. Simple heuristics are employed to define boundaries between the individual repeat elements. Diagonal distances that are in good agreement define the element length and sequences of this length from hit end-points provide representative elements of the array.

PILER-TR finds families of elements with "Terminal Repeats". This search is geared towards finding transposable elements with terminal repeats, such as the long LTR retrotransposons. The signature of these features is a set of repeats, about 50–2000 bp, separated by anywhere from 50 to 15 000 bp (all default parameters). A banded search is used to find candidate terminal repeats. To avoid reporting tandem repeats and pseudo-satellites that would also be found via a banded search, these are found and masked as a prerequisite to this search. After finding candidate terminal repeats, a second search is carried out to find different elements with matching terminal repeats, in which case a nonbanded regular search is performed. All candidates with matching terminal repeats are clustered and reported as families of terminal repeat elements.

2.7 RepeatScout

Alkes Price and Pavel Pevzner's RepeatScout [46] takes a distinct approach to repeat family identification, which circumvents some of the difficulties associated with the more traditional approach involving the postprocessing of pairwise alignments. Generating pairwise alignments as the first step of repeat sequence identification can take a long time, utilize many CPU cycles and generate copious output that can consume an enormous amount of disk space. With large genomes, this can be intractable and further effort is required to partition data sets into more manageable inputs, all of which can adversely affect the results obtained. In contrast, RepeatScout employs a repeat family search stage heuristic similar to that used in database-searching algorithms like BLASTN [1b]. Where BLASTN requires that two potentially homologous sequences share at least one exact word match in common, RepeatScout requires that all initially targeted members of a repeat family share at least one exact word in common. In basic terms, RepeatScout uses an exact word match (seed) to identify potential members of a repeat family, and then maximally extends alignments of all targeted regions to the left and right of the seed to compute a consensus sequence representation of the repeat family with repeat boundaries optimized. The specifics of this approach are described below.

The first phase of the RepeatScout algorithm involves scanning the genome for frequently occurring words. A collection of genomic sequences are scanned and the positions of all words of user specified length (i.e. 13-mers) are catalogued. Closely spaced repeat word occurrences are ignored to avoid tandem repeats [tandem repeat finding is relegated to Tandem Repeat Finder (TRF) [9] and is not an objective of RepeatScout]. After the scanning is complete, frequently occurring word matches are fed to the final phase of RepeatScout – the repeat family identification and consensus sequence construction stage.

Starting with the most frequently occurring word, RepeatScout attempts to extend all such word occurrences to the left and to the right, terminating the extension at what are considered to be the most appropriate boundaries of the repeated element, and simultaneously generating a consensus sequence for this repeat family. The extension phase is perhaps the most distinctive and critical feature of the RepeatScout algorithm, and it is the extension algorithm that rigorously defines the repeat boundaries. The consensus sequence generated by the word extension phase is optimally aligned to all members of that repeat family and cannot be further extended without reducing the total alignment score. This is accomplished by the following objective function, which computes the score of the consensus sequence as the sum of the scores of each individual repeat element aligned to the consensus:

$$A(Q;S_1,\ldots,S_n) = \left\lfloor \sum_k \max\{a(Q,S_k),0\} \right\rfloor - c * \text{length}(Q)$$

where $a(Q,S_k)$ corresponds to the score of an alignment between the consensus sequence Q and a genome sequence substring (S_k) of equal length that extends from both ends of the seed. The constant c imposes a minimum threshold on the number of individual repeat elements (S_n) that must align with the consensus sequence to provide a suitable representation of a repeat family.

The choice of alignment function $a(Q,S_k)$ determines how the consensus sequence boundaries are positioned. With a Smith–Waterman local alignment function [54], short partial repeats would not be penalized and spurious alignment extensions to the more complete elements could drive the consensus boundary position beyond more appropriate repeat boundaries. Towards the other extreme, a fit-alignment algorithm, which fits one sequence into another [57] could be used to force all underlying complete and partial elements to match the consensus, but this can have the affect of yielding consensus boundaries that underrepresent the true boundaries. As a more suitable compromise between these two scenarios, the authors introduce a *fit-preferred* alignment function that yields a consensus sequence shared by some but not all of the underlying complete and partial copies. The fit-preferred alignment function is described below:

$$\begin{aligned}
f(i,0) &= \max(-\gamma i, -p)\ , \\
f(0,j) &= 0\ , \\
f(i,j) &= \max \begin{cases} f(i-1,j-1) + \mu_{ij} \\ f(i,j-1) - \gamma \\ f(i-1,j) - \gamma \\ -p \end{cases} , \\
a(Q,S) &= \max_{i,j} \begin{cases} f(i,j) & \text{if} \quad i = |Q| \\ f(i,j) - p & \text{if} \quad i < |Q| \end{cases}
\end{aligned}$$

where the match/mismatch score is provided by μ_{ij}, the gap penalty score by $\gamma\gamma$ and the fixed incomplete-fit penalty provided by p. Here, $f(i,j)$ is the score of a best alignment between the $1, \ldots, i$ characters in the consensus sequence Q and $1, \ldots, j$ characters in the repeat element S. The fit-preferred alignment score $a(Q,S)$ is simply $f(i,j)$ if the best alignment includes the entire consensus sequence. If not, the incomplete-fit penalty is subtracted from the best alignment score, penalizing the alignment for not including the entire consensus sequence.

The fit-preferred alignment algorithm is used by RepeatScout to generate alignments separately to the left and to the right of the word match. The fixed incomplete-fit penalty (p) is subtracted from the score of any optimal alignment to the consensus sequence that fails to extend all the way to the left boundary of the consensus sequence; an analagous penalty applies to the right boundary. If the alignment is incomplete on both boundaries, the penalty is subtracted twice. The result is that partial copies of the element are penalized in the presence of longer, more complete elements, and the consensus sequences that are generated are more suitable representations of the underlying repeat copies targeted by the exact word match. False-positive candidate elements targeted by the initial word matching strategy do not pose problems for RepeatScout; these will acquire negative alignment scores and are eliminated from contributing to the consensus by virtue of the main objective function.

The most rigorous approach to generating the consensus sequence would involve n-dimensional dynamic programming (where n is the number of sequences), but this would not be practical or even possible for more than a few sequences given that this task would be NP-hard. Instead, a heuristic approach is taken to generate the consensus whereby the word match is extended to the left and right one nucleotide at a time. A single nucleotide extension is attempted using each of the four nucleotides (G, A, T and C) and the single nucleotide extension providing the optimal alignment score is chosen. The consensus sequence is constructed greedily in this way until a maximal score is obtained and a predetermined number of subsequent iterations fails to improve upon this maximal score. The consensus sequence providing the maximal score is chosen to represent the underlying set of repeats and the termini of the consensus sequence delimit the repeat boundaries.

The RepeatScout algorithm, as described, is applied to each frequently occurring word match, beginning with those most frequent. As a single repeat family is likely to contain many exact word matches, effort must be taken to prevent re-identifying the same repeat family based on other yet-to-be processed frequently occurring words. In an attempt to prevent this effect, the counts of words found within approximate occurrences of the consensus sequence are readjusted within the set of frequently occurring words, decreasing the chance but not absolutely preventing the possibility of finding the same (or a portion of a) repeat family identified previously. A future release of RepeatScout may improve upon this functionality for identifying approximate occurrences of the consensus sequence, in order to more completely preclude repeat family rediscovery based on subsequent word matches.

The task of finding occurrences of repeat family members in the genome is relegated to searching the genome with the database of consensus sequences using RepeatMasker, BLAST or another sequence search and alignment utility. These homology searches may find repeat elements that have diverged considerably from those used by RepeatScout to generate the consensus (i.e. they lacked an exact word match required to be included in the consensus sequence construction stage of RepeatScout). This procedure provides a powerful mechanism to rigorously identify and annotate the individual elements of a larger repeat family.

3 Tandem Repeats

Tandem repeats form a special class of repetitive sequences, composed of a contiguous stretch of two or more copies of a repeat pattern. The length of the repeat pattern is called the period. This class of repeats, termed satellites, is of great biological relevance, found to correspond to specialized structures within eukaryotic genomes such as the short pentamer to heptamer repeats that form telomeres, and the longer repeats that form centromeres (e.g. 180-bp period repeats in *Arabidopsis*). Microsatellites (SSRs), found both within and between genes, are of great interest to those studying biodiversity and population genetics, and for DNA fingerprinting studies. Expansions in trinucleotide repeats have been correlated with various disease states, including Huntington's disease and Friedreich's ataxia, among others.

The problem of finding tandem repeats has received much attention from computer scientists and biologists alike, due to both the tractability of the problem from an algorithmic perspective and because of the importance of tandem repeats in their diverse biological roles. The challenge of finding tandem repeats involves identification of the repeated pattern and the number of times the pattern is repeated. Over the past decade, many algorithms have been proposed for the identification of tandem repeats, some of which seem to be academic exercises in algorithm development and few are found implemented in publicly available software for the general application of tandem repeat finding in the postgenomic era. One exception is Gary Benson's TRF [9], which has seen widespread use in genome sequence analysis and remains the most popular tandem repeat analysis software today. Alternative tools for finding tandem repeats have recently become available and extend the repertoire of essential software available to genome researchers. Here, we survey a few of these tools and describe the algorithms employed for finding tandem repeats.

3.1 TRF

Gary Benson's TRF is a powerful software tool capable of finding exact and approximate (containing mismatches or indels) tandem repeats in genomic sequences [9]. The algorithm employed in TRF is broken into two stages; first, the detection component which identifies candidate tandem repeats using a set of statistical criteria, followed by the analysis component calculates the consensus repeat pattern and period size using sequence alignment methods.

TRF initially detects tandem repeats based on the premise that similar sequences found contiguously arrayed are likely to share exact substrings, and the distance between paired substrings will be approximately the same and correspond to the period of the tandem repeat. In a search for these exact substrings that may target a tandem repeat, the genomic sequence is scanned from left to right for exact word matches, with some fixed word length w.

All further analyses of the candidate tandem repeat during this detection phase rely on statistical analyses whereby the alignment between candidate tandem repeats are modeled as a sequence of independent and identically distributed (iid) Bernoulli trials, equivalent to a sequence of coin tosses such that heads correspond to matching pairs of nucleotides, and tails correspond to mismatches or indels. The probability pM of a match and the probability pI of an indel are user-defined parameters that provide an upper limit to the allowed divergence between candidate tandem repeats.

The probabilistic model of the iid Bernoulli sequence is used with several statistical criteria to evaluate candidate tandem repeats: the *sum of heads distribution* to dictate the number of matches required among candidate repeats; the *random walk distribution* to model the indels between tandem repeats that might cause variability in the apparent period length; the *apparent size distribution* to distinguish between tandem repeats and dispersed repeats by analyzing the distribution of matches along the proposed period length; and the *waiting time distribution* to choose match search criteria that are most suitable for different period lengths. Each of the above distributions depend on the period length, word length used for scanning matches, and the user-defined cutoffs of pM and pI.

During the scan of the genomic sequence from left to right, word matches are accumulated. The position of each word is kept in a history list and the distance between word matches is kept in a distance list. Once a word match is found, the distance separating the words is presumed a candidate period length for a tandem repeat. The candidate tandem repeat would require additional matches along the remainder of its period length. These additional matches are found by querying the distance list, searching for word matches separated by the same period length, with the leading word match positioned between the triggering word matches. The statistical test using the sum of heads distribution determines the minimum number of matches required for a candidate tandem repeat. Here, the normal distribution is used to determine the minimum number x of matches, such that 95% of the time, at least x nucleotides (heads) are counted as part of exact word matches (head runs of length w) along the period length. To account for indels between word pairs of approximate tandem repeats, the period length is not fixed at a constant during this phase, but allowed to vary slightly, consistent with the random walk distribution. The allowed variation is restricted to that expected within 95% of random occurrences based on the indel probability *pI*, under the hypothesis of a random walk along one dimension with maximal displacement equal to *d***pI*.

Candidate tandem repeats which pass the sum of heads test are further analyzed to differentiate tandem repeats from local repeats that are not arrayed in tandem. Tandem repeats are distinguished from nontandem direct repeats, i.e. repeats found in close proximity but not directly adjacent, by the distribution of matches along the period length of the candidate tandem repeat. Nontandem direct repeats will tend to have matches concentrated on the right side of the period length (because the algorithm processes them from left to right), whereas the tandem repeats should have leading word matches distributed throughout. The apparent repeat length is calculated as the maximal distance between the first and last run of matches found using the exact word match scan. This apparent repeat length is likely to be smaller than the actual repeat length, but provides a useful approximation for this analysis. A minimum apparent repeat length threshold for a tandem repeat is determined by simulation. An apparent size distribution is generated from random Bernoulli sequences using the *pM* value to model an alignment between two genuine tandem repeats with period length *d*, and the distances between the first and last runs of exact word matches are collected. A minimum apparent repeat length is chosen such that 95% of the time, the apparent repeat length determined for random Bernoulli sequences with *pM* exceeds this cutoff length. Candidate tandem repeats passing the minimum apparent repeat length threshold are further subjected to the analysis component.

The waiting time distribution is used to pick word lengths used during the initial genome scan. Random Bernoulli sequences are used to determine the minimum number of aligned residues (coin tosses) to find an exact word match (run of heads) of length w, 95% of the time, given a probability of a match (*pM* or probability of heads). As with other sequence alignment software, such as BLAST, the choice of word length is very important and affects the sensitivity and running time of the analysis. Short word lengths accumulate large history lists and many false-positive matches, unlikely to be indicative of tandem repeats. Alternatively, large word lengths accumulate few false positives, but are unlikely to detect short approximate tandem repeats. Therefore, the word length needs to be chosen in accordance with the repeat length under consideration. The waiting time distribution is used to pick a set of word lengths to apply to different ranges of pattern sizes, given *pM*. Word lengths of 3–7 bp are chosen to detect tandem repeats with periods of up to 500 bp and at least 75–80% identity.

Those candidate tandem repeats passing the above statistical criteria are further examined under the analysis component of TRF. The analysis component involves aligning the interval of the candidate tandem repeat to the surrounding genomic sequence using the technique of wrap-around dynamic programming (WDP) [18], nicely described in Ref. [10] and Appendix A of Ref. [8]. The technique of WDP provides a method whereby a single copy of the tandem repeat can be aligned with all copies in a larger stretch of genomic sequence, such that the alignment is allowed to wrap around the tandem repeat from end to beginning again, to continue alignments to the subsequent copies of the repeat. The candidate tandem repeat used in WDP may not be the optimal sequence, as the consensus among the repeat copies may contain nucleotide differences or indels when compared to the initial candidate used to generate the alignments. A consensus pattern is generated from the alignment, and this consensus is realigned to calculate the period length and number of copies of the tandem repeat. TRF is limited to finding tandem repeats with unit lengths up to 500 bp.

3.2 STRING (Search for Tandem Repeats IN Genomes)

The algorithm underlying STRING relies almost exclusively on sequence alignment methods to identify tandem repeats [40]. As with TRF, the algorithm consists of two stages; first, the identification of candidate tandem repeats, followed by a more detailed analysis stage to resolve the tandem repeat structures. The identification of candidate tandem repeats involves what are referred to as autoalignments, which involves aligning a sequence to itself. A variation of the Waterman–Eggert algorithm [58] is implemented to identify all nonintersecting local alignments, with a modification to avoid reporting the trivial alignment of the complete sequence to itself along its entirety. Features of some autoalignments are found characteristic of tandem repeats: aligned sequence pairs with overlapping or neighboring coordinates indicate a tandem repeat with a period equal to the distance between coordinates of aligned residues. Such autoalignments do not rigorously define the tandem repeat, but rather highlight the regions of genomic sequences which are strong candidates for containing tandem repeats, to be analyzed in a subsequent tandem repeat finding search stage. Candidate regions are selected by grouping all autoalignments with overlapping coordinates and including those nonoverlapping autoalignments that are found in close proximity that could be extensions of tandem repeats not captured during the autoalignment stage.

Each candidate region is further subjected to the tandem repeat search stage, as follows. A set of words are chosen such that each distinct word is a potential isolated element of larger tandem repeat. A variation of WDP is used to align each word against the larger candidate region. This involves performing the Waterman–Eggert-style alignment using a cyclically addressed word, capturing all high scoring non-intersecting local alignments. Each word and alignment, as a unit, is referred to as a Single-Expansion Interpretative Pattern (SIP). All SIPs found within a candidate region are compared to each other in a pairwise fashion in order to eliminate redundancy and resolve conflicts between overlapping SIPs (some SIPs may be found as insignificant versions of larger SIPs). Remaining SIPs are reported as tandem repeat tracts with the triggering word as the consensus for the tandemly repeated element. STRING is limited to finding tandem repeats of unit length smaller than 100 bp.

3.3 MREPS

A distinguishing feature of the newer MREPS program is its ability to find tandem repeats of any length, from microsatellites to large tandem segmental genome duplications [30]. At the heart of MREPS is a very efficient combinatorial algorithm based on advanced string processing techniques, which finds approximate tandem repeats, also called k-mismatch repeats, running in linear time $O(nk\log(k) + S)$ for a sequence of length n containing S repeats with at most k mismatches per tandem repeat copy. MREPS finds all k-mismatch repeats for values of k up to a user-specified maximal resolution parameter, enabling the program it to find highly divergent repeats. Additional processing time is spent refining the results of this search to report biologically meaningful repeats, coping with artifacts resulting from the algorithmic definition of k-mismatch repeats and consolidating redundant repeats found at different k-mismatch runs, as described below.

The mathematical definition of the k-mismatch repeat requires that the repeat be maximal. For this purpose, mismatches are sometimes added to the repeat boundaries, extending the repeat length to enforce the maximal k-mismatch repeat definition. MREPS attempts to identify these unwanted extensions and trim them from the repeat termini, retaining the more meaningful and longer core of the repeat.

Another postprocessing step consolidates redundancy among repeats and computes their optimal period value. The same region of genomic sequence can be reported as having tandem repeat sequences with different periods. For example, a tandem repeat with period of 2 may also be reported with periods that are multiples of 2. Each period may be associated with a different degree of degeneracy, based on the k-mismatch limit for reporting the repeat. The optimal period for the repeat is chosen as that which minimizes the number of mismatches between tandem copies with period p.

This is done for every period from 1 to p and the period with minimum error rate is reported. After computing the optimal period, repeats with the same period and overlapping by at least two periods are merged together to form a single repeat. By doing this, repeats originally found as k-mismatch repeats are redefined in a more satisfying way.

The tandem repeats found as a result of this process are filtered to retain only those that are considered to be statistically significant. In this case, the statistically significant repeats are those that are unlikely to be found within random sequences. Empirical thresholds for minimum length and maximum error rates were determined for various resolution parameters using shuffled genomic sequences, and these thresholds are applied to the collection of repeats to remove insignificant entries.

4 Repeats and Genome Assembly Algorithms

Genome assembly is perhaps the computationally most demanding task in genomics, requiring days or weeks of computation time for the largest genomes, even on the latest vintage computers. The assembly problem itself is simple enough to state: given a collection of input sequences, compute how these sequences overlap one another and use these overlaps to reconstruct the original chromosomes. A large mammalian genome assembly generated from a WGS sequencing project might include over 20 million input sequences of approximately 800 bp in length. Most of the sequences are generated in pairs, by sequencing both ends of a larger DNA fragment; these fragments are grouped into “libraries” with a characteristic fragment size. The assembly algorithm must keep track of those sizes in order to place the sequence pairs (or “mates”) approximately the right distance apart in the final assembly. A thorough account of the computational assembly of genomes is given in Chapter 2.

Repetitive sequences make genome sequence assembly hard; without repeats, almost any algorithm can correctly assembly a genome. This follows from the fact that without repeats, any overlapping sequence shared by two or more individual sequence “reads” clearly implies that the reads came from the same chromosomal location and can be assembled together. As repeats are so central to the assembly problem, much effort has been dedicated within large-scale assembly systems to the repeat identification problem.

Assembly systems are only looking for repeats that will confuse them, which are a subclass of all repeats. First of all, assembly algorithms must compare large numbers of reads (as many as 30 million) looking for overlaps. The fundamental goal of assembly from a WGS sequencing project is to unify overlapping reads if they originated from the same place on the same chromosome. Consequently, the reads should be identical up to the limits of sequencing error. Thus, assemblers look for shared sequences that are nearly identical – a typical threshold is to require that two reads must overlap by at least 40 bp and the overlapping region must be at least 98% identical. This leads naturally to the observation that any pair of repetitive sequences that is less than 98% identical will not cause any serious problems for assembly. Such divergent repeats can be sorted out and placed into the correct locations in the genome. Of course, this is a somewhat simplistic view; e.g. it is often the case that short regions in the middle of long repeats are identical – and therefore confusing – even if the entire repeat is not.

Second, any repeat that is contained entirely within a sequencing read does not cause a problem, because the unique sequence flanking the repeat will allow the read to be placed correctly in the genome assembly. Current sequencing technology generates reads of 800 bp or longer; therefore a repeat region that spans less than 800 bp rarely presents a problem. Note that the phrase "repeat region" here refers both to single-copy repeats and to tandem repeats. If a repeat occurs in 20-bp units, but those units occur in tandem arrays spanning 100 copies, then the repeat region spans 2000 bp and is definitely a problem for assembly, even though the repeat unit itself is quite short.

Thus, it should be clear that assembly algorithms must identify sequence reads that are comprised entirely of repetitive sequence, and they must handle these repeat reads differently. For the sake of discussion, we will describe how they are handled in the Celera Assembler [38], although many of these strategies are similar to those employed by Arachne [5,25] and other current assemblers. (Note that the Celera Assembler is now open source and includes many enhancements not described in the original paper; the code is available at http://sourceforge.net/projects/wgs-assembler.)

4.1 Repeat Management in the Celera Assembler and other Assemblers

There are two main tasks in repeat processing for assembly: (i) one must identify repeats and (ii) one must attempt to place them in the assembled genome. We will discuss these two issues in order.

4.2 Repeat Identification by k-mer Counts

The first major computation in most assembly algorithms is the overlap step, in which all reads must be compared to all other reads. In order to compute this essentially quadratic operation efficiently, most assemblers employ a hashing strategy: they create a hash table and record in it all *k*-mers of a certain length. Each *k*-mer entry stores the read identifier and the position within that read where the *k*-mer occurred. Typical values of *k* are 22 (used in the TIGR Assembler [55] and Celera Assembler) and 24 (used in the Arachne assembler); this is long enough that in a random DNA sequence, the vast majority of *k*-mers will not occur at all.

The *k*-mer hash array provides a simple and natural vehicle for identifying repeats. Recall that we want to identify sequence reads that are entirely repetitive; i.e. satisfying the condition that no unique sequence can be found within the read. After scanning all reads for all *k*-mers, it is a trivial matter to note the average depth of coverage by computing the mean number of entries for each *k*-mer in the table. Based on this value, one can determine a threshold above which a *k*-mer can safely be assumed to be repetitive, e.g. 3 times the mean. The Celera Assembler then scans the reads a second time and for each read looks at the counts of each *k*-mer in the read. If all *k*-mers in a read have a count above the threshold for repeats, then the read itself is labeled as a repeat. Arachne takes a slightly different approach, eliminating all *k*-mers that are overrepresented so that they will not be used in the overlap calculation.

4.3 Repeat Identification by Depth of Coverage (Arrival Rates)

A second method for identifying repeats occurs later during the assembly process, after at least one round of contig creation. Once the assembler has a set of contigs built, it can ask whether entire contigs are repetitive. The simplest method here is based on coverage: for a genome covered at, for example, 8 times coverage, any contig with significantly deeper coverage is highly likely to be repetitive. As the average coverage is easy to compute, it is also easy to detect any contigs whose coverage is 2 or 3 times normal. However, such a simple approach fails to account for the fact that, statistically, a longer contig is likely to have coverage closer to the mean than a short contig. For example, in an 8 times assembly, a long contig with 15 times coverage is far more likely to represent a repeat than is a short contig with the same coverage.

The Celera Assembler models the expected coverage of a contig using an "arrival rate" statistic. The idea is the following: assuming that the WGS reads are generated by a uniform random process that samples every location in the genome equally, then the reads should "arrive" at a contig (i.e. they should align to it) at a rate that can be modeled as a Poisson process. This arrival rate statistic is computed as follows [38]. Suppose that the genome size is G, the sequencing project generated F reads and we are examining a contig containing k of those reads. Consider the positions where all k reads begin in the contig; these are the arrival locations. If the contig occurs just once in the genome, then we should have sampled it at the same rate as any other interval on the genome; in this case, if we look at the region of length r between the first and last arrival locations, then the probability of seeing $k - 1$ arrivals in that interval is $[(rF/G)^k/k!]e^{(-rF/G)}$. If the contig occurs twice in the genome (and therefore we expect twice the arrival rate) then the probability of seeing $k - 1$ arrivals is $[(2rF/G)^k/k!]e^{(-2rF/G)}$. The "$A$ statistic" in the Celera Assembler is computed as the log ratio of these two probabilities; i.e. $A = (\log e)rF/G - (\log 2)k$. In simple terms, this statistic computes whether a contig is more likely to occur once than twice in the genome. By adjusting this single parameter, the assembler can be more or less cautious about what it considers a repeat.

4.4 Repeat Identification by Conflicting Links

A third way of identifying repeats is to notice that a contig has two or more adjacent contigs according to mate-pair information. If a repeat occurs at multiple distinct locations (i.e. not in tandem), then the paired sequences that align to the repeat will have mates (links) that point to different loci. This can be recognized relatively easily during the scaffolding stage of assembly.

4.5 Repeat Placement: Rocks and Stones

Once a repeat has been identified, the assembler must decide what to do with it. A number of assemblers, including Celera Assembler, first assemble the obviously nonrepetitive sequences and then try to place the repeats. One strategy for doing this in Celera Assembler is called the "rocks and stones" approach. The idea is to first build scaffolds from the unique sequences, linking contigs together only if at least two mate-pairs (i.e. the pair of sequences that comes from opposite ends of a single DNA fragment) agree on the linkage. Rocks and stones are repetitive contigs (based on the A statistic) that are "thrown" into these gaps if mate-pair links indicate they belong there; rocks must have at least two links and stones only need one. The assembler then attempts to join the flanking contigs together by finding a tiling of reads across the contigs and the newly placed stones. Note that one weakness of this approach is that it is sometimes difficult to determine whether one or multiple copies of a stone belong in a gap and it is possible that the wrong number of copies of a tandem repeat will end up in the assembly.

4.6 Repeat Placement: Surrogates

One final issue surrounding repeats in an assembly is the precise mapping of the reads to the final consensus sequence. Even if the consensus sequence, i.e. the genome sequence that ends up being deposited in public archives, is correct, it may be impossible to determine exactly which sequences belong to particular regions of the genome. For regions that are 100% identical and that are longer than a single read, multiple reads can be mapped to multiple distinct genomic locations and it may simply be impossible to tell where each read goes, since all the repetitive reads can go in each copy of the repeat. The Celera Assembler algorithm handles this problem through the use of "surrogate" contigs: the reads are assembled into a contig which is labeled as a surrogate, meaning that it apparently occurs more than once in the genome. This surrogate can then be placed in multiple locations in the genome based on mate pairs and on reads that span the repeat boundary. However, internal reads cannot be mapped to the genome, so the final consensus assembly points only to the surrogate, but not to the multi-alignment of all the reads. An alternative strategy, used in Arachne, is to place the reads multiply, allowing them to occur at two or more locations in the genome. Neither solution is entirely satisfactory, but in both cases the genome assembly can be constructed correctly.

4.7 Repeat Resolution in Euler

A different approach to repeat identification is taken by the Euler assembler [45]. In this unusual assembly algorithm, the normal overlap computation is handled quite differently from the hash table approach of Celera Assembler and Arachne. Instead, an overlap graph is created in which nodes represent overlap and edges represent *k*-mers. Contigs can be created by finding an Eulerian path through such a graph, called a de Bruijn graph, a problem that can be solved in linear time. (A Eulerian path through a graph is a path that uses each edge in the graph exactly once, here meaning that each overlap is realized exactly once – from left to right – as it should be in assembling genomes.) To reconstruct the sequence of the contig, the algorithm follows the Euler path and "reads off" the *k*-mers found on each edge in the path.

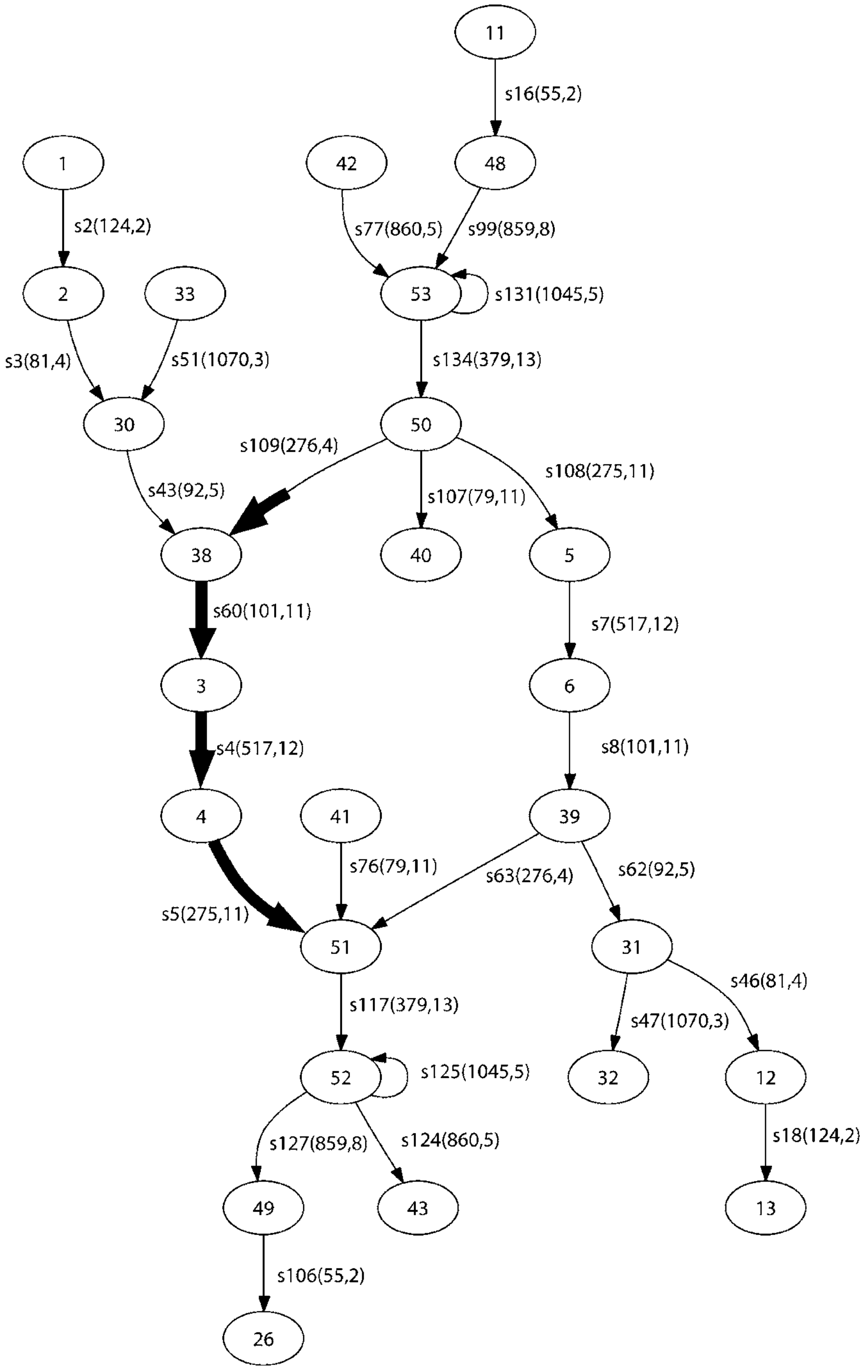
A fuller description of the Euler assembler is beyond the scope of this review, but it is worth mentioning how it handles repeats. In the de Bruijn graph created for the purpose of assembly, repeats appear as edges that share the same label. These edges can be superposed, yielding a new data structure (the A-Bruijn graph, see below) in which some nodes have many edges entering or exiting them. Then repetitive sequences correspond to edges whose boundaries are at such nodes. A critical aspect of the Euler algorithm is its error correction: in any large genome project, the small number of sequencing errors in individual reads can easily be confused with slight variations between repeat copies. By reducing the error rate, the algorithm can more easily tell if two near-identical sequences represent two different repeat copies. Euler takes advantage of the fact that errors tend to be random and, therefore, that short *k*-mers with a very low count probably represent such errors. This enables the algorithm to identify and correct a large majority of sequencing errors, and this in turn enables it to separate repeats that are very close to identical.

Despite all the efforts in Celera Assembler, Arachne, Euler and other assemblers, some classes of repeats continue to confound large-scale genome assembly algorithms. Although this is not widely known or discussed, the genomes available today in public archives likely contain numerous assembly errors. The most common errors are collapses of tandem repeats into too few copies; in addition, gross rearrangements around repeats have also been discovered (and in some cases corrected). The continuing problems highlight the fact that more work needs to be done to continue to improve the quality of "finished" genomes.

5 Untangling the Mosaic Nature of Repeats (The A-Bruijn Graph)

Pavel Pevzner and coworkers introduce a graph data structure, the A-Bruijn graph, to describe repeated sequences ascertained from pairwise alignments and to reveal the complex mosaic structure commonly encountered among related sequences [44]. This A-Bruijn graph has found use in several applications, including genome fragment assembly, multiple sequence alignment and *de novo* repeat finding [44, 48]. The graph glues similar sequence regions together into edges of the graph. When applied to a single genome, the glued-together edges correspond to repeated sequences, and when applied to related sequence sets from different genomes, homologous sequence regions are found glued together. The graph itself provides a compact view of the related regions among a collection of sequences, in addition to the sequence regions found to be unique to a single sequence or subset of sequences. Software packages implementing the A-Bruijn graph include the ABA multiple sequence aligner [48], and the *de novo* repeat-finding program RepeatGluer [44], which is the focus here.

The A-Bruijn graph is constructed from a set of pairwise alignments. All genome alignments generated by an alignment program (those tested include BLAST, PatternHunter and BLAT, among others) are decomposed into the A-Bruijn graph by "gluing" paired genome regions together as edges in a graph bounded by nodes, with similar sequences forming a single edge and forking at a node into separate edges where sequences diverge. After constructing the graph, edges with a multiplicity greater than one correspond to repeated regions of sequences. By removing all nonrepetitive sequences from the graph by discarding all edges with multiplicity of one, the A-Bruijn graph is broken into sets of connected components termed tangles. The tangles represent


11
s16(55,2)
1
42
48
s2(124,2)
s77(860,5)
s99(859,8)
2
33
53
s131(1045,5)
s3(81,4)
s51(1070,3)
s134(379,13)
30
50
s109(276,4)
s108(275,11)
s43(92,5)
s107(79,11)
38
40
5
s60(101,11)
s7(517,12)
3
6
s4(517,12)
s8(101,11)
4
41
39
s76(79,11)
s63(276,4)
s62(92,5)
s5(275,11)
51
31
s46(81,4)
s117(379,13)
s47(1070,3)
52
s125(1045,5)
32
12
s127(859,8)
s124(860,5)
s18(124,2)
49
43
13
s106(55,2)
26

the repeat elements, specifically the structure of subrepeats that form larger mosaic repeats. Each complete repeat element found in the genome can be reconstructed by traversing edges of valid paths in the corresponding tangle, and the relationships among different mosaic repeats are elucidated by virtue of their subrepeats found in common.

The identity and structure of each representative repeat element, termed by Pevzner and coworkers as the "elementary repeat", is provided by the A-Bruijn graph as a maximal simple path with multiplicity greater than one. RepeatGluer generates the repeat graph in a text file format compatible with graph viewing software (i.e. dotty program of the Graphviz package [17]). Additionally, the genomic sequences corresponding to each subrepeat are extracted and a consensus for each is provided in FASTA format. An example of a repeat graph is shown in Figure 4, highlighting the largest repeat tangle found in the repeat-rich genome of *Deinococcus radiodurans*.

6 Repeat Annotation in Genomes

Given the great diversity and functional significance of repetitive sequences, their annotation in completed genomes is an important task. Annotating repeats is similar to annotating other features in the genome in that a set of coordinates is required to delimit the feature location, along with a description of the biological significance of that feature, if known. The location of repetitive sequences can be obtained in two different ways or in their combination. Repeats are mined directly from the genome sequence based on algorithms that locate repeated sequences based exclusively on the genome sequence composition; these tools and algorithms were the focus of previous

Figure 4 Largest tangle in the repeat graph for *Deinococcus radiodurans* as constructed by RepeatGluer. Repeat consensus sequences yielded for edges in this tangle mostly correspond to parts of transposons (often called "insertion sequences" in bacterial genomics). The coding sequence for the transposase of one transposon (annotated gene identifier DRB0020, gi10957398) is shown threaded through the graph with wide edges. This graph was illustrated using graphviz. Unique identifiers are assigned to each node and edge. Edges correspond to subrepeats and nodes bound the subrepeats that are found in common between larger repeats. The length of the subrepeat and its multiplicity are specified to the right of the edge identifier. Only edges with multiplicity above 1 are shown. As repeats on the forward strand can be merged into single edges with repeats on the reverse strand, the graph is constructed using both strands as if they were independent sequences, which yields symmetry in the resulting graph; the transposase coding sequence described above is highlighted on only one side of the symmetric graph.

discussions. Alternatively, repeats can be identified based on homology to entries in a preexisting library of known repeat sequences.

The advantage of the former method is that no prior knowledge is required. Any newly sequence genome can be applied to the earlier described *de novo* repeat finding methods to rapidly identify the repeats. These are considered *de novo* methods simply because repeats are found based on an analysis of the genome sequence alone, without prior knowledge of the location or sequence composition of the repeats. The disadvantage is that other than knowing that a sequence is a repeat, we do not know what the repeat sequence represents biologically (i.e. gene, transposon, satellite, segmental duplication).

The latter method, based on repeat libraries, is currently the most widely used and most trusted method for annotating repeat sequences. Entries of repeat libraries are typically well annotated with some indication of function when functional information is known, and homology found to a known repeat reveals both its location and identity. Both software and repeat libraries are available as resources for genome annotation. Repbase began as a collection of human representative repeat sequences and fragments, and grew into a large collection of repeats from a variety of (mostly) model organisms, now known as Repbase Update [27]. Repbase Update includes separate repeat libraries for primates, rodents, zebrafish, *C. elegans*, *Drosophila* and *Arabidopsis*, with prototype sequences that correspond to consensus of large families and subfamilies of repeats. A "Simple" library, which can be applied to any genome, provides entries that help identify low complexity microsatellite sequences.

The program originally used to search genome sequences against Repbase libraries is CENSOR [28]. Due to the growing data volume in the libraries and the need for faster searching programs, CENSOR was eventually replaced by the more efficient alignment program RepeatMasker [53]. RepeatMasker identifies regions of homology to Repbase entries using Phil Green's cross_match algorithm (http://www.genome.washington.edu/UWGC/analysistools/Swat.cfm) and then replaces these homologous regions in the genomic sequence with "N" characters, effectively masking them in the genome sequence. A further 30-fold speed increase is obtained by using WU-BLAST in place of cross_match, as implemented by MaskerAid [6] as an enhancement to RepeatMasker. By masking the sequence, these repeat regions are precluded from subsequent sequence analyses in a larger annotation pipeline; hidden from gene finding programs, and transcript and protein homology searches, focusing subsequent analyses on the remaining unique sequence. For perspective, almost half of the human genome is masked due to the repeat content and this step is incredibly important when trying to find components of genes that remain hidden in the unmasked nonrepetitive regions.

The disadvantage with the repeat library-based scanning method is that comprehensive repeat libraries are available only for those organisms that have been well studied, and have attained the status of model organisms. Repbase is a tremendous resource for repeat annotation when corresponding repeat libraries are available, but it is of limited utility for many organisms whose genomes are currently being sequenced because of the limits of homology detection at the level of the nucleotide sequence coupled with the rate of divergence within repeats across evolutionary boundaries. Independent efforts are sometimes necessary to generate comprehensive repeat libraries to supplement that offered by Repbase [39].

A use of *de novo* repeat-finding programs is to automate the generation of repeat libraries that can be used with RepeatMasker. RepeatScout, in particular, yields consensus sequences for repeat families that are to be subsequently searched against the genome using RepeatMasker to identify locations of individual members of the family. Care must be taken with this approach if the repeats are to be masked from the genomic sequence and hidden from subsequent analyses. The Repbase libraries include many transposable elements and other repeat features which are excluded from the host gene set, and so masking homologous features from the genome should not interfere with subsequent efforts to find genes. The output from *de novo* repeat finding tools contains transposable elements but may also include repetitive features such as members of large gene families, and by blindly masking these "repeat" features from the genome, important features will be inadvertently disguised. None of the *de novo* repeat-finding programs directly address the problem of deducing the biological significance of the repeats that are found computationally. This is a difficult problem, currently left to the biologists and bioinformaticians, and examined on a case-by-case basis.

Searching for occurrences of repeats using representative repeat sequences or consensus sequences is limited by the information provided by that single sequence. By searching with profile representations of repeat families, the sensitivity of a search can be improved; a study by Juretic and coworkers demonstrates improved sensitivity in the detection of transposable elements in Rice by using hidden Markov model (HMM) profiles created for known transposable element families [26]. Although this methodology or similar profile methods are popular for finding members of protein families or occurrences of protein domains, HMM profiles for repeats are not currently widely employed, but do show great promise for repeat detection and analysis, and should be considered along with the existing alternatives.

Acknowledgments

We would like to thank the following individuals for reviewing an earlier version of the manuscript: Stefan Kurtz, Pavel Pevzner, Bob Edgar, Zhirong Bao, and Alkes Price. Your comments were invaluable. S.L.S. was supported in part by the US National Institutes of Health under grants R01-LM06845 and R01-LM007938.

Repeat Finding Tools and Resources

Miropeats and ICAtools	http://www.littlest.co.uk/software/bioinf/index
mreps	http://mreps.loria.fr
PILER	http://www.drive5.com/piler
RECON	http://www.genetics.wustl.edu/eddy/recon
Repbase Update	http://www.girinst.org/Repbase_Update
RepeatFinder	http://www.tigr.org/software
RepeatGluer	http://nbcr.sdsc.edu/euler/intro_tmp
RepeatMasker	http://www.repeatmasker.org
RepeatScout	http://www-cse.ucsd.edu/groups/bioinformatics/software
REPuter	http://bibiserv.techfak.uni-bielefeld.de/reputer
STRING	http://www.caspur.it/∼castri/STRING/index.htm.old
Tandem Repeat Finder	http://tandem.bu.edu/trf/trf

References

1 **a.** Agarwal, P., D. J. States. 1994. The Repeat Pattern Toolkit (RPT): analyzing the structure and evlution of the *C. elegans* genome. Proc. Int. Conf. Intell. Syst. Mol. Biol. **2**: 1–9.
b. Altschul, S. F., W. Gish, W. Miller, E. W. Myers and D. J. Lipman. 1990. Basic local alignment search tool. J. Mol. Biol. **215**: 403–10.

2 Altschul, S. F., T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**: 3389–402.

3 Arumuganathan, K. and E. D. Earle. 1991. Nuclear DNA content of some important plant species. Plant Mol. Biol. Rep. **9**: 208–218.

4 Bao, Z. and S. R. Eddy. 2002. Automated *de novo* identification of repeat sequence families in sequenced genomes. Genome Res. **12**: 1269–76.

5 Batzoglou, S., D. B. Jaffe, K. Stanley, et al. 2002. ARACHNE: a whole-genome shotgun assembler. Genome Res. **12**: 177–89.

6 BEDELL, J. A., I. KORF AND W. GISH. 2000. MaskerAid: a performance enhancement to RepeatMasker. Bioinformatics **16**: 1040–1.

7 BENNETZEN, J. L., P. SANMIGUEL, M. CHEN, A. TIKHONOV, M. FRANCKI AND Z. AVRAMOVA. 1998. Grass genomes. Proc. Natl Acad. Sci. USA **95**: 1975–8.

8 BENSON, G. 1997. Sequence alignment with tandem duplication. J. Comput. Biol. **4**: 351–67.

9 BENSON, G. 1999. Tandem repeats finder: a program to analyze DNA sequences. Nucleic Acids Res. **27**: 573–80.

10 BENSON, G. AND M. S. WATERMAN. 1994. A method for fast database search for all k-nucleotide repeats. Nucleic Acids Res. **22**: 4828–36.

11 CAPY, P., C. BAZIN, D. HIGUET, AND T. LANGIN. 1998. *Dynamics and Evolution of Transposable Elements*. Landes Bioscience, Austin, TX.

12 CHOO, K. H., B. VISSEL, A. NAGY, E. EARLE AND P. KALITSIS. 1991. A survey of the genomic distribution of alpha satellite DNA on all the human chromosomes, and derivation of a new consensus sequence. Nucleic Acids Res. **19**: 1179–82.

13 CRAIG, N. L. 2002. *Mobile DNA II*. ASM Press, Washington, DC.

14 DELCHER, A. L., S. KASIF, R. D. FLEISCHMANN, J. PETERSON, O. WHITE AND S. L. SALZBERG. 1999. Alignment of whole genomes. Nucleic Acids Res. **27**: 2369–76.

15 DELCHER, A. L., A. PHILLIPPY, J. CARLTON AND S. L. SALZBERG. 2002. Fast algorithms for large-scale genome alignment and comparison. Nucleic Acids Res. **30**: 2478–83.

16 EDGAR, R. C. AND E. W. MYERS. 2005. PILER: identification and classification of genomic repeats. Bioinformatics **21** (Suppl. 1): i152–8.

17 ELLSON, J., E. GANSER, Y. KOREN, et al. 2005. Graphviz – Graph Visualization Software. http://www.graphviz.org/.

18 FISCHETTI, V., G. LANDAU, J. SCHMIDT AND P. SELLERS. 1992. Identifying periodic occurrences of a template with applications to protein structure (presented at the 3rd Annual Symposium on Combinatorial Pattern Matching). Lecture Notes Comput. Sci. **644**: 111–120.

19 FLAVELL, R. B., M. D. BENNETT, J. B. SMITH AND D. B. SMITH. 1974. Genome size and the proportion of repeated nucleotide sequence DNA in plants. Biochem. Genet. **12**: 257–69.

20 FRASER, C. M., S. CASJENS, W. M. HUANG, et al. 1997. Genomic sequence of a Lyme disease spirochaete, *Borrelia burgdorferi*. Nature **390**: 580–6.

21 GISH, W. AND D. J. STATES. 1993. Identification of protein coding regions by database similarity search. Nat. Genet. **3**: 266–72.

22 GUSFIELD, D. 1997. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press, Cambridge;

23 HESLOP-HARRISON, J. S., M. MURATA, Y. OGURA, T. SCHWARZACHER AND F. MOTOYOSHI. 1999. Polymorphisms and genomic organization of repetitive DNA from centromeric regions of *Arabidopsis* chromosomes. Plant Cell **11**: 31–42.

24 HUANG, X. AND W. MILLER. 1991. A time-efficient, linear-space local similarity algorithm. Adv. Appl. Math. **12**: 337–357.

25 JAFFE, D. B., J. BUTLER, S. GNERRE, E. MAUCELI, K. LINDBLAD-TOH, J. P. MESIROV, M. C. ZODY AND E. S. LANDER. 2003. Whole-genome sequence assembly for Mammalian genomes: arachne 2. Genome Res. **13**: 91–6.

26 JURETIC, N., T. E. BUREAU AND R. M. BRUSKIEWICH. 2004. Transposable element annotation of the rice genome. Bioinformatics **20**: 155–60.

27 JURKA, J. 2000. Repbase update: a database and an electronic journal of repetitive elements. Trends Genet. **16**: 418–20.

28 JURKA, J., P. KLONOWSKI, V. DAGMAN AND P. PELTON. 1996. CENSOR – a program for identification and elimination of repetitive elements from DNA sequences. Comput. Chem. **20**: 119–21.

29 KENT, W. J. 2002. BLAT – the BLAST-like alignment tool. Genome Res. **12**: 656–64.

30 KOLPAKOV, R., G. BANA AND G. KUCHEROV. 2003. mreps: efficient and

flexible detection of tandem repeats in DNA. Nucleic Acids Res. **31**: 3672–8.

31 KURTZ, S., J. V. CHOUDHURI, E. OHLEBUSCH, C. SCHLEIERMACHER, J. STOYE AND R. GIEGERICH. 2001. REPuter: the manifold applications of repeat analysis on a genomic scale. Nucleic Acids Res. **29**: 4633–42.

32 KURTZ, S., A. PHILLIPPY, A. L. DELCHER, M. SMOOT, M. SHUMWAY, C. ANTONESCU AND S. L. SALZBERG. 2004. Versatile and open software for comparing large genomes. Genome Biol. **5**: R12.

33 KURTZ, S. AND C. SCHLEIERMACHER. 1999. REPuter: fast computation of maximal repeats in complete genomes. Bioinformatics **15**: 426–7.

34 LANDER, E. S., L. M. LINTON, B. BIRREN, et al. 2001. Initial sequencing and analysis of the human genome. Nature **409**: 860–921.

35 LI, M., B. MA, D. KISMAN AND J. TROMP. 2004. Patternhunter II: highly sensitive and fast homology search. J. Bioinform. Comput. Biol. **2**: 417–39.

36 MA, B., J. TROMP AND M. LI. 2002. PatternHunter: faster and more sensitive homology search. Bioinformatics **18**: 440–5.

37 MARGOLIS, R. L., M. G. MCINNIS, A. ROSENBLATT AND C. A. ROSS. 1999. Trinucleotide repeat expansion and neuropsychiatric disease. Arch. Gen. Psychiatry **56**: 1019–31.

38 MYERS, E. W., G. G. SUTTON, A. L. DELCHER, et al. 2000. A whole-genome assembly of *Drosophila*. Science **287**: 2196–204.

39 OUYANG, S. AND C. R. BUELL. 2004. The TIGR Plant Repeat Databases: a collective resource for the identification of repetitive sequences in plants. Nucleic Acids Res. **32**: D360–3.

40 PARISI, V., V. DE FONZO AND F. ALUFFI-PENTINI. 2003. STRING: finding tandem repeats in DNA sequences. Bioinformatics **19**: 1733–8.

41 PARSONS, J. D. 1995. Improved tools for DNA comparison and clustering. Comput. Appl. Biosci. **11**: 603–13.

42 PARSONS, J. D. 1995. Miropeats: graphical DNA sequence comparisons. Comput. Appl. Biosci. **11**: 615–9.

43 PEARSON, W. R. AND D. J. LIPMAN. 1988. Improved tools for biological sequence comparison. Proc. Natl Acad. Sci. USA **85**: 2444–8.

44 PEVZNER, P. A., H. TANG AND G. TESLER. 2004. *De novo* repeat classification and fragment assembly. Genome Res. **14**: 1786–96.

45 PEVZNER, P. A., H. TANG AND M. S. WATERMAN. 2001. An Eulerian path approach to DNA fragment assembly. Proc. Natl Acad. Sci. USA **98**: 9748–53.

46 PRICE, A. L. AND P. A. PEVZNER. 2005. *De novo* identification of repeat families in large genomes. Bioinformatics.

47 PROKOPOWICH, C. D., T. R. GREGORY AND T. J. CREASE. 2003. The correlation between rDNA copy number and genome size in eukaryotes. Genome **46**: 48–50.

48 RAPHAEL, B., D. ZHI, H. TANG AND P. PEVZNER. 2004. A novel method for multiple alignment of sequences with repeated and shuffled elements. Genome Res. **14**: 2336–46.

49 READ, T. D., S. L. SALZBERG, M. POP, et al. 2002. Comparative genome sequencing for discovery of novel polymorphisms in *Bacillus anthracis*. Science **296**: 2028–33.

50 RICHARDS, R. I. AND G. R. SUTHERLAND. 1997. Dynamic mutation: possible mechanisms and significance in human disease. Trends Biochem. Sci. **22**: 432–6.

51 RICHARDS, R. I. AND G. R. SUTHERLAND. 1996. Repeat offenders: simple repeat sequences and complex genetic problems. Hum. Mutat. **8**: 1–7.

52 ROMERO, D., J. MARTINEZ-SALAZAR, E. ORTIZ, C. RODRIGUEZ AND E. VALENCIA-MORALES. 1999. Repeated sequences in bacterial chromosomes and plasmids: a glimpse from sequenced genomes. Res. Microbiol **150**: 735–43.

53 SMIT, A. F., R. HUBLEY AND P. GREEN. 1996–2004. RepeatMsker Open-3.0.

54 SMITH, T. F. AND M. S. WATERMAN. 1981. Identification of common molecular subsequences. J. Mol. Biol. **147**: 195–7.

55 SUTTON, G., O. WHITE, M. ADAMS AND A. R. KERLAVAGE. 1995. TIGR Assembler: a new tool for assembling large shotgun sequencing projects. Genome Sci. Technol. **1**: 9–19.

56 VOLFOVSKY, N., B. J. HAAS AND S. L. SALZBERG. 2001. A clustering method for repeat analysis in DNA sequences. Genome Biol. **2**: RESEARCH0027.

57 WATERMAN, M. S. 2000. *Introduction to Computational Biology: Maps, Sequences And Genomes: Interdisciplinary Statistics*. Chapman & Hall/CRC Press, Boca Raton, FL.

58 WATERMAN, M. S. AND M. EGGERT. 1987. A new algorithm for best subsequence alignments with application to tRNA–rRNA comparisons. J. Mol. Biol. 197: 723–8.

8 Analyzing Genome Rearrangements

Guillaume Bourque

1 Introduction

The study of comparative maps and the rearrangements they evidence was pioneered in the late 1910s at the Morgan *Drosophila* lab [45, 74]. In the context of phylogenetics, the analysis of genome rearrangements was first introduced by Dobzhansky and Sturtevant in a study of inversions in *Drosophila pseudoobscura* [22]. What followed was a succession of developments in the fields of comparative mapping and comparative genomics. In particular, breakthroughs in mapping and sequencing afforded genome-wide analyses of gene order in various sets of genomes [3, 12, 20, 27, 51, 53, 54, 63]. Recently, the considerable investments in large sequencing projects have made accessible detailed sequences and maps for many eukaryotic genomes [26, 32, 37, 79, 84]. One of the stated purpose of these endeavors is to further our understanding of these species through comparative analyses [50]. The availability of these large genomes leads to great opportunities, but also challenges, in the study of genome rearrangements.

The exploration of large-scale events shaping whole-genome architecture provides a complementary perspective on the evolution of these organisms as compared to more traditional molecular studies focused on the analysis of individual genes. In fact, rearrangement studies allow detailed reconstructions of evolutionary scenarios, including ancestral reconstructions of entire eukaryotic genomes [13, 14]. Furthermore, such analyses can lead to the identification of regions of genomic instability (high rates of rearrangements, breakpoint reuse, etc.) that challenge and help refine our understanding of the dynamics of chromosome evolution [46, 57]. A related problem, also associated with genomic instability, is the study of cancer. Rearrangements in a tumor genome can be analyzed very much as if the tumor was a new organism that had recently diverged from the normal human genome. The interest is that although cancer progression is frequently associated with genome rearrangements, the forces behind these alterations are still poorly understood.

Bioinformatics - From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

This chapter is organized as follows. Section 2 presents some of the basic concepts required for the analysis of genome rearrangements such as how the genomes are modeled and what types of rearrangements are considered. Section 3 presents three criteria that can be used to compute the distance between a pair of genomes: the breakpoint distance, the rearrangement distance and the conservation distance. Section 4 shows how the same criteria can be use to infer phylogenies when multiple genomes are considered using three different approaches: distance-based, maximum parsimony and maximum likelihood. Section 5 presents a few recent applications of analysis of genome rearrangements in large genomes and also recent work studying genome rearrangements in cancer. Finally, Section 6 concludes with some remarks on important challenges and promising new developments for the comparative analyses of gene order.

2 Basic Concepts

2.1 Genome Representation

Initially, the focus of genome rearrangement studies was on the comparative analyses of small genomes such as mitochondria [12, 53, 54, 63], chloroplasts [20, 51, 54], viruses [27] and small region of larger genomes [3]. In this context, the relative order of homologous genes in different organisms was used to infer phylogenetic relationships and even rearrangement scenarios. An example showing differences between the order of homologous genes in two mitochondria is given in Figure 1.

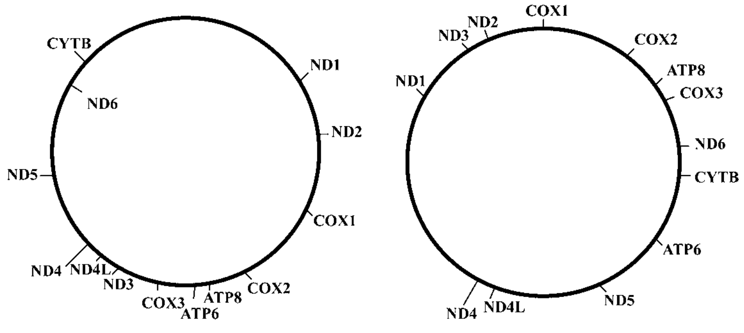


Figure 1 Coding genes on the human and on the earthworm mitochondrial DNA (mtDNA). The list of genes is the same, but their order differs. For instance, ND1 and ND2 are adjacent in Human but they are seperated by ND3 in the earthworm. tRNA genes have been left out of this figure to simplify the example. GenBank accession numbers: NC_001807 and NC_001673.

For this purpose, the relative gene order of different genomes can be encapsulated into a set of *signed permutations*. One of the genomes is identified as the reference genome and is associated to the identity permutation where each integer corresponds to one of its genes. The permutation associated to each other genome can directly be obtained from the order of appearance of the homologous genes. Furthermore, a sign corresponding to the relative orientation (strand) of the gene, as compared to the reference genome, is given to each integer of the new permutation. To continue with the example shown in Figure 1, if the mtDNA is selected as the reference genome, and we label the genes starting with COX1 in Human as "1" and going clockwise until ND2 is assigned "13", we obtain the permutations shown in Table 1. Of all the genes in the two mtDNAs, only ND6 in Human was on the reverse DNA strand, which is why it is represented by "-10" in Earthworm as its relative orientation is reversed.

Table 1 Signed permutations associated with the two mitochondria genomes shown in Figure 1

Human	1	2	3	4	5	6	7	8	9	10	11	12	13
Earthworm	1	2	3	5	-10	11	4	9	7	8	12	6	13

This genome representation can be adapted and generalized for data sets with other distinctive features such as multiple chromosomes, unsigned gene orders, unequal gene content and different source of homology markers. We briefly present these variants.

2.1.1 Circular, Linear and Multichromosomal Genomes

A genome can consist of a single chromosome or a collection of chromosomes and is called *unichromosomal* or *multichromosomal* accordingly. There are two types of chromosomes: circular and linear. The mitochondria shown in Figure 1 are circular genomes. Linear chromosomes have two separated endpoints. Unless otherwise stated, multichromosomal genomes will be assumed to have linear chromosomes. The different types of genomes can also be represented by permutations, but additional markers are required for multichromosomal genomes to mark the boundaries of the chromosomes.

It is important to specify the type of chromosomes we are considering because they will lead to different equivalent representations. For instance, consider the three genomes:

$$\begin{array}{rcrrr} G_1 & = & 1 & 2 & 3 \\ G_2 & = & 2 & 3 & 1 \\ G_3 & = & -3 & -2 & -1 \end{array}$$

As circular chromosomes, all three representation are equivalent; however, as linear chromosomes, only G_1 and G_3 correspond to the same representation (there is usually no distinction between the two end-points of a linear chromosome and so a complete flip leads to an equivalent representation).

2.1.2 Unsigned Genomes

If the orientation of genes in a set of genomes is unknown, the relative gene order can still be encapsulated into a set of unsigned permutations. Unfortunately, many genome-rearrangement problems, such as calculating the reversal distance, are significantly harder for unsigned permutations [17] compared to signed permutations (see Section 3.2). For this reason, but also because the relative orientation is usually obtainable, we will assume that we are dealing with signed genomes in the rest of this chapter.

2.1.3 Unequal Gene Content

In representing genomes with permutations, we have assumed that a set of n genes was found with a unique homologous counterpart in all genomes. In fact, in many cases this assumption will be violated: a genome may have gained additional genes through rearrangements events such as insertions or duplications and it may have lost genes following deletions. To encapsulate the relative gene orders of genomes with unequal gene content we need to generalize the representation to account for this variable alphabet. Although models that are not restricted to equal gene content are more complete and realistic (see Ref. [24] for a review), they are also more challenging algorithmically and have been limited to few applications [23, 66, 69]. We will focus on genomes with equal gene content in the rest of this chapter except for the presentation of some rearrangement events affecting gene content in Section 2.2.

2.1.4 Homology Markers

So far, it was implicit that the signed permutations representing the genomes were constructed based on the relative position of homologous genes. Actually, however, any type of marker with an homologous counterpart in all genomes can be used to construct similar permutations. This is important because, especially in large eukaryotic genomes (e.g. human, mouse) where genes only cover a small fraction of the genome, the ability to use markers extracted from raw DNA sequence allows to study rearrangement events that occur anywhere in the genome. This will be covered in more detail in Section 5.1.

2.2 Types of Genome Rearrangements

During evolution, an assortment of events can modify the genome. These events are known as mutations and they can occur especially during DNA replication. Mutations are divided into two major categories: point mutations and chromosomal mutations. Point mutations are at the single-base level. Although point mutations can have a significant impact on the genome (e.g. a base change could be responsible to the insertion of an early stop codon that completely annihilates a gene), they will not be considered further here as they mostly affect individual genes. In contrast, chromosomal mutations affect directly the architecture of genomes by modifying the gene order or the gene content. There are various types of chromosomal mutations, but the most common are reversals (or inversions), translocations, fusions, fissions, transpositions, inverted transpositions, insertions, deletions and duplications. See Figure 2 for a cartoon example of how reversals and deletions could occur during DNA replication. Only translocations, fusions and fissions are specific to multichromosomal genomes. The first six types of chromosomal mutations rearrange the genes, but they do not modify the set of genes present in a given genome. Insertions, deletions and duplications, on the other hand, modify the gene content of a genome by adding, or removing, some genes or by generating multiple copies of the same gene. The effect of these different chromosomal mutations are exemplified in Table 2.

Table 2 Examples of chromosomal mutations that impact either gene order or gene content

Mutation type	Before		After	Impact
Reversal	$1\ 2\ \boxed{3\ 4\ 5}\ 6$	⇒	$1\ 2\ -5\ -4\ -3\ 6$	gene order
Translocation	$1\ 2\ \boxed{3\ 4\ 5}$ $6\ 7\ \boxed{8}$	⇒	1 2 8 6 7 3 4 5	gene order
Fusion	$\boxed{1\ 2\ 3\ 4}$ $\boxed{5\ 6}$	⇒	1 2 3 4 5 6	gene order
Fission	$\boxed{1\ 2\ 3\ 4}\ \boxed{5\ 6}$	⇒	1 2 3 4 5 6	gene order
Transposition	$1\ \boxed{2\ 3}\ 4\ 5\ \boxed{}\ 6$	⇒	1 4 5 2 3 6	gene order
Inverted transposition	$1\ \boxed{2\ 3}\ 4\ 5\ \boxed{}\ 6$	⇒	$1\ 4\ 5\ -3\ -2\ 6$	gene order
Insertion	$1\ 2\ 3\ 4\ \boxed{}\ 5\ 6$	⇒	1 2 3 4 7 5 6	gene content
Deletion	$1\ \boxed{2\ 3}\ 4\ 5\ 6$	⇒	1 4 5 6	gene content
Duplication	$1\ 2\ \boxed{3\ 4}\ 5\ 6$	⇒	$1\ 2\ 3\ 4\ 3'\ 4'\ 5\ 6$	gene content

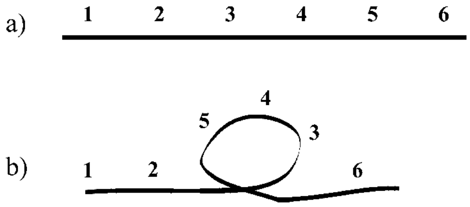


Figure 2 (a) DNA fragment with each integer corresponding to a gene. (b) The same DNA fragment but twisted. During replication, if the twisted loop is copied, it leads to a *reversal* and the fragment becomes 1 2 −5 −4 −3 6. Note that the sign or the strand of the genes is modified at the same time as the order. If the twisted loop is ignored, it results in a *deletion* and the fragment is transformed into 1 2 6.

3 Distance between Two Genomes

In this section, we review different criteria that can be use to measure the distance between two genomes based on comparative gene order. The first criterion, the *breakpoint distance*, counts the number of disruptions of the relative gene order between a pair of genomes. The second criterion, the *rearrangement distance*, relies on the *a priori* definition of a set of permissible operations (e.g. only reversals) and then minimizes the number of such operations required to convert one gene order into the next. The final criterion described is the *conservation distance*, which, similar to the breakpoint distance, circumvents the requirement of a rearrangement model. Under this criterion, the disruption of the relative gene order is measured by the number of conserved or common intervals.

3.1 Breakpoint Distance

The breakpoint distance [48,85] compares two permutations by directly counting the number of gene order disruptions between two genomes. Formally, given two signed permutations of size n, π and γ, the first step to compute the breakpoint distance is to extend both permutations so that they start with 0 and end with $n+1$: $\pi = 0, \pi_1, \pi_2 \ldots \pi_n, n+1$ and $\gamma = 0, \gamma_1, \gamma_2 \ldots \gamma_n, n+1$. Then, the *breakpoint distance*, $d_{\text{break}}(\pi, \gamma)$, is defined as the number of pairs (γ_i, γ_{i+1}), $0 \leq i \leq n$, such that neither the pair (γ_i, γ_{i+1}) nor $(-\gamma_{i+1}, -\gamma_i)$ appears in π. For instance, using the example from Table 1 and setting $\pi =$ human and $\gamma =$ earthworm, we get $d_{\text{break}}(\pi, \gamma) = 9$. The nine breakpoints are displayed in γ using arrows:

$$\begin{array}{ccccccccccccccccccccccccccccc} 0&&1&&2&&3&&5&&-10&&11&&4&&9&&7&&8&&12&&6&&13&&14\\ &&&&&&&\uparrow&&\uparrow&&\uparrow&&\uparrow&&\uparrow&&\uparrow&&&&\uparrow&&\uparrow&&\uparrow&&& \end{array}$$

Two important strengths of this criterion measuring the degree of similarity are that (i) it is easily computable in linear time and (ii) it does not require any assumptions about the underlying rearrangement mechanisms.

3.2 Rearrangement Distance

Given two permutations π and γ and a set of permissible rearrangements, the rearrangement distance, $d_{\text{rear}}(\pi, \gamma)$, is defined as the minimum number of operations required to convert one permutation into the other. For example, given that reversals are the only allowed operations, what is the minimum number of events required to convert the permutation associated with the earthworm mtDNA into the one associated with the human mtDNA shown in Table 1? The problem is quite challenging. In this particular case, the answer is seven and Table 3 shows one such scenario. We will use d_{rev} for the special case of d_{rear} when reversals are the only permissible operations.

The interest in looking for the minimum number of steps is that, under the assumption that such events are rare (and that our rearrangement model is correct), we hope to recover the sequence of rearrangements that really occurred. The caveat is that it is well known that the most parsimonious scenario underestimates the actual number of operations when this number is above a threshold of θn, where n is the size of the permutation and θ is in the range from $1/3$ to $2/3$ [15,35,83].

Table 3 Example of a most parsimonious rearrangement scenario with seven reversals between earthworm and human mtDNA

Earthworm	1	2	3	5	−10	11	4	9	7	8	12	6	13
$\rho(6,11)$	1	2	3	5	−10	11	4	9	7	8	12	6	13
$\rho(9,12)$	1	2	3	5	−10	−12	−8	−7	−9	−4	−11	6	13
$\rho(7,10)$	1	2	3	5	−10	−12	−8	−7	−6	11	4	9	13
$\rho(6,12)$	1	2	3	5	−10	−12	−11	6	7	8	4	9	13
$\rho(4,6)$	1	2	3	5	−10	−9	−4	−8	−7	−6	11	12	13
$\rho(4,7)$	1	2	3	9	10	−5	−4	−8	−7	−6	11	12	13
$\rho(6,10)$	1	2	3	4	5	−10	−9	−8	−7	−6	11	12	13
Human	1	2	3	4	5	6	7	8	9	10	11	12	13

Based on different sets of permissible rearrangements, various methods have been proposed to efficiently compute the rearrangement distance and sort a pair of genomes. Of all the choices of permissible operations, the reversal-only model is probably the most extensively studied. The work was pioneered by Sankoff and Kececioglu [68], but was followed by the development of increasingly efficient polynomial-time algorithms [1,8,9,28,34]. Other studied sets of permissible operations include transpositions [2,81], inversions, translocations, fusions and fissions [29, 52, 75], and more recently block interchange (a more general type of transposition) [19, 41, 86].

In the remainder of this section, we review a methodology that was developed to compute the distance between a pair of genomes using reversals only (for unichromosomal genomes) or reversals, translocations, fusions and fissions (for multichromosomal genomes). We will refer to it as the Hannenhalli–Pevzner (HP) theory. This methodology was developed in Bafna and Pevzner [4] and in Hannenhalli and Pevzner [30], it was summarized in Pevzner [58], it was improved in Tesler [75], and, finally, it was implemented in a program called GRIMM [76].

3.2.1 **HP Theory**

We first describe the methodology for unichromosomal genomes where reversals are the only permissible operations. Assume we have a permutation γ that we wish to sort with respect to the identity permutation π. The first step is to convert γ, a signed permutation, into γ', an unsigned permutation, by mimicking every directed element i by two undirected elements i^{t} and i^{h} representing the tail and the head of i. Since γ is a permutation of size n, γ' will be a permutation of size $2n$. We now extend the permutation γ' by adding $\gamma'_0 = 0$ and $\gamma'_{2n+1} = n + 1$. The next step is to construct the breakpoint graph associated with γ. The *breakpoint graph* of γ, $G(\gamma)$, is an edge-colored graph with $2n + 2$ vertices. Black edges are added between vertices γ'_{2i} and γ'_{2i+1} for $0 \leq i \leq n$. Grey edges are added between i^{h} and $(i+1)^{\mathrm{t}}$ for $0 < i < n$, between 0 and 1^t, and between n^h and $n + 1$. Black edges correspond to the actual state of the permutation while grey edges correspond to the sorted permutation we seek. See Figure 3 for an example.

Bafna and Pevzner [4], and later Hannenhalli and Pevzner [30], showed that $G(\gamma)$ contains all the necessary information for efficiently sorting the permutation γ. The first step is to look at the maximal cycle decomposition of the breakpoint graph. Finding the maximal cycle decomposition of a graph in general can be a very difficult problem; however, fortunately, because of

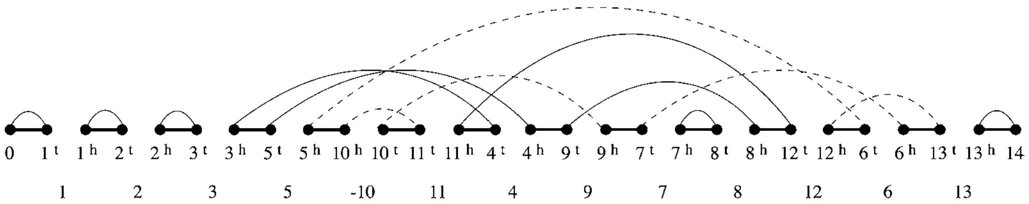


Figure 3 Breakpoint graph associated with the two permutations from Table 1. Black edges are shown using think lines. All other lines (both solid and dashed) correspond to grey edges. Dashed lines are used to show the only nontrivial oriented cycle.

the way the breakpoint graph was constructed for a signed permutation, each vertex has degree two and so the problem is trivial. Suppose $c(\gamma)$ is the maximum number of edge-disjoint alternating cycles in $G(\gamma)$. The cycles are alternating because, in the breakpoint graph of a signed permutation, each pair of consecutive edges always has different colors.The important lower bound:

$$d_{\text{rev}}(\pi,\gamma) = d(\gamma) \geq n + 1 - c(\gamma).$$

was first presented by Kececioglu and Sankoff [35].

A few additional concepts on the breakpoint graph are required to present the result of Hannenhalli and Pevzner [28]. A grey edge in $G(\gamma)$ is said to be *oriented* if it spans an odd number of vertices (when the vertices of $G(\gamma)$ are arranged in the canonical order $\gamma'_0, \ldots, \gamma'_{2n+1}$). A cycle is said to be *oriented* if it contains at least one oriented grey edge. Cycles which are not oriented are said to be *unoriented* unless they are of size 2, in which case they are said to be *trivial*. The term "oriented" comes from the fact that if we traverse an oriented cycle we will traverse at least one black edge from left to right and one black edge from right to left. In the breakpoint graph shown in Figure 3, there are only two nontrivial cycles: one where the grey edges are displayed using solid lines and one where the grey edges are displayed using dashed lines. The cycle with solid lines is unoriented since it does not contain an oriented edge but the cycle with dashed lines is oriented because it contains an oriented edge [e.g. $(10^{\text{h}}, 11^{\text{t}})$].

For each grey edge in $G(\gamma)$ we will now create a vertex v_e in the *overlap graph*, $O(G(\gamma))$. Whenever two grey edges e and e$'$ overlap or cross in the canonical representation of $G(\gamma)$, we will connect the corresponding vertices v_{e} and $v_{\text{e}'}$. A *component* will mean a connected component in $O(G(\gamma))$. A component will be oriented if it contains a vertex v_e for which the corresponding grey edge e is oriented. As for cycles, a component which consists of a single vertex (grey edge) will be said to be trivial. In Figure 3, there are five trivial components and one larger oriented component since at least one of its grey edge is oriented. The difficulty in sorting permutations comes from unoriented components.

Unoriented components can be classified into two categories: hurdles and protected nonhurdles. A *protected nonhurdle* is an unoriented component that separates other unoriented components in $G(\gamma)$ when vertices in $G(\gamma)$ are placed in canonical order. A *hurdle* is any unoriented component that is not a protected nonhurdle. A hurdle is a *superhurdle* if deleting it would transform a protected nonhurdle into a hurdle, otherwise it is said to be a *simple hurdle*. Finally, γ is said to be a *fortress* if there exists an odd number of hurdles and all are superhurdles in $O(G(\gamma))$ [71].

The main result fromHannenhalli and Pevzner [28] is that:

$$d_{\mathrm{rev}}(\pi,\gamma) = d(\gamma) = n + 1 - c(\gamma) + h(\gamma) + f(\gamma),$$

where $h(\gamma)$ is the number of hurdles in γ, and $f(\gamma)$ is 1 if γ is a fortress and 0 otherwise. However, the machinery to recover an optimal sequence of sorting reversals was also presented. The fact that the distance between the human and earthworm is 7 can directly be extracted from this formula and from the breakpoint graph shown in Figure 3 since there are 13 genes, seven cycles and no hurdles or fortress.

Finally, Hannenhalli and Pevzner [29] derived a related equation to compute the rearrangement distance between two multichromosomal genomes when permissible operations are: reversals, translocations, fusions and fissions. We refer the reader to Pevzner [58] and Tesler [75] for the details of the calculation, but we will briefly present how the formula can be obtained.

The main idea to compute the rearrangement distance between two multichromosomal genomes Π and Γ is to concatenate their chromosomes into two permutations π and γ. The purpose of these concatenated genomes is that every rearrangement in a multichromosomal genome Γ can be mimicked by a reversal in a permutation γ. In an *optimal* concatenate, sorting γ with respect to π actually corresponds to sorting Γ with respect to Π. Tesler [75] also showed that when such an optimal concatenate does not exist, a *near-optimal* concatenate exists such that sorting this concatenate mimics sorting the multichromosomal genomes and uses a single extra reversal which corresponds to a reordering of the chromosomes.

3.3 Conservation Distance

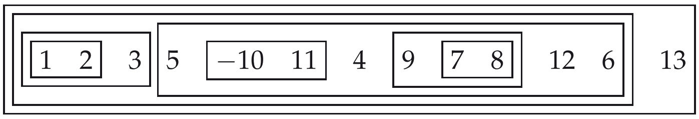
Recently, two criteria were proposed to measure the level of similarity between sets of genomes: common intervals [31,78] and conserved intervals [7]. In a way, both of these criteria represent a generalization of the breakpoint distance but consider intervals instead of only adjacencies. There are two important properties that common/conserved distances share with the breakpoint distance:

(i) It can be directly defined on a set of more than two genomes and allows the identification of shared features in a family of organisms.

(ii) It does not rely on an a priori model of rearrangements.

3.3.1 Common Intervals

Given two signed permutations, π and γ, a *common interval* is a set of two or more integers that is an interval in both permutations [31,78]. Using the example from Table 1, we get that there are 14 common intervals, eight of which are shown in the earthworm using boxes:



The additional common intervals not displayed are: $[2,3]$, $[2,3,5,\ldots 6]$, $[2,3,5,\ldots 13]$, $[3,5,\ldots 6]$, $[3,5,\ldots 13]$ and $[5,10,\ldots 13]$.

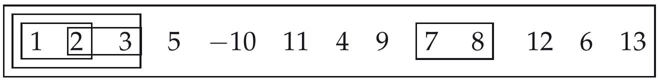
Suppose $C(\pi,\gamma)$ and $C_i(\pi,\gamma)$ are the number of common intervals and the number of common intervals of size i in π and γ, respectively. We note that the maximum number of common intervals for two permutations of size n is achieved for identical permutations and is simply:

$$\sum_{i=2}^{n} C_i(\pi,\pi) = \quad (n-1)+(n-2)+\ldots+1 = \frac{n(n-1)}{2}.$$

Of course, the more common intervals between two permutations, the higher the conservation. In the example above, there is only 14 common intervals while the maximum achievable is 78.

3.3.2 Conserved Intervals

Given two permutations, π and γ, a *conserved interval* is an interval $[a,b]$ such that a precedes b or $-b$ precedes $-a$, in both π and γ, and the set of elements, without signs, between a and b is the same in both π and γ [7]. Continuing with the example from Table 1, there are only five conserved intervals between the human and earthworm mtDNA:



Although, initially the definition of conserved intervals may seem unnatural, it is tightly connected to the HP theory (it corresponds to *subpermutations* in Ref. [29]) and it was shown that it can be used to efficiently sort permutations by reversals [5].

4 Genome Rearrangement Phylogenies

An important challenge in the comparative analysis of gene order is the construction of phylogenies based on genome rearrangements that describe the genetic relationships between the organisms. Phylogenies are represented by unrooted binary trees such that the leaf nodes of the trees correspond to contemporary genomes and the internal nodes correspond to their extinct ancestors (see Figure 5 for an example). Phylogenetic tree reconstruction is difficult largely because the number of unrooted trees grows at a rate that is more than exponential with the number of leaf nodes.

We review three main classes of approaches that can be used for phylogenetic tree reconstruction based on relative gene order: distance-based methods, maximum parsimony methods and maximum likelihood methods. These main classes of approaches are very similar in spirit to the ones developed for phylogenetic tree reconstruction based on sequence evolution with point mutations instead of chromosomal mutations (see Chapter 4). Links for some of the programs available to analyze genome rearrangements described in this section are provided in Table 4.

Table 4 Links for some of the software tools available to analyze genome rearrangements

BPAnalysis	http://www.cs.washington.edu/homes/blanchem/software
GRAPPA	http://www.cs.unm.edu/∼moret/GRAPPA
GOTREE	http://www.mcb.mcgill.ca/∼bryant/GoTree
GRIMM	http://www-cse.ucsd.edu/groups/bioinformatics/GRIMM
MGR	http://www-cse.ucsd.edu/groups/bioinformatics/MGR
BADGER	http://badger.duq.edu

4.1 Distance-based Methods

These approaches construct trees strictly based on the pairwise distances between the leaf nodes of the tree. The first step computes the pairwise distance matrix for the genomes of interest using one of the criterion described in Section 3 or from other criterion such as EDE, the “empirically derived estimator", that attempts to correct the bias in the parsimony assumption for large distances [83]. Distance-based methods differ in the second step in how they make use of the distance matrix to reconstruct the trees. Currently, the most common family of distance-based methods is probably “neighbor-joining” which was first proposed by Saitou and Nei [62].

Methods in this class are typically very efficient; in many cases phylogenies can be inferred in polynomial time. When applied to gene order data, one of the limitations of distance-based approaches is that they do not label the internal nodes and they do not associate a rearrangement scenario to the phylogeny. For challenging data sets, this may lead to infeasible or less accurate solutions [82]. This limitation is addressed both by maximum parsimony and by maximum-likelihood methods.

4.2 Maximum Parsimony Methods

Methods seeking the most parsimonious scenario attempt to recover the tree, and its internal nodes, that minimizes the number of events on its branches. It corresponds to the Steiner Tree Problem [33] on various metrics. The first methods of this type were developed for sequence data [25] but they were later adapted for gene order data [27,64]. Formally, given a set of m genomes, the problem is to find an unrooted tree T, where the m genomes are leaf nodes, and assign internal ancestral nodes such that $D(T)$ is minimized:

$$D(T) = \sum_{(\pi,\gamma)\in T} d(\pi,\gamma),$$

where $d(\pi,\gamma)$ can be any of the distances described in Section 3. The special case of three genomes ($m = 3$) is called the median problem. Although the tree topology for this problem is trivial, the assignment of the optimal internal node can still be challenging.

If a rearrangement distance is used, a detailed rearrangement scenario could also be associated to the tree that will describe every intermediate step of the evolution of these genomes. Again, under the assumption that rearrangements events are rare [15,61], its reasonable to seek the most parsimonious scenario to recover the actual tree.

Although many of the pairwise distances can be computed in polynomial time (e.g. the breakpoint distance d_{break} and the reversal distance d_{rev}, see Section 3.2), it was shown that both the median problem for d_{break} and the median problem for d_{rev} are NP-hard [16,18,55]. Nevertheless, there are a few efficient heuristics to tackle both the median problem [67,72] and the full phylogeny problem [10,15,44] under different sets of assumptions. We briefly present some of these methods.

Sankoff and Blanchette [67] studied the median problem for the breakpoint distance; they described a clever reduction of this problem to the Traveling Salesman Problem for which reasonably efficient algorithms are available. Using this result, Blanchette and coworkers [10] developed BPAnalysis, a method to recover the most parsimonious scenario for m genomes also under the breakpoint distance. That method first looked for the optimal assignment of internal nodes for a given topology by solving a series of median problem (this is also known as the small parsimony problem). The next step was to scan the space of all possible tree topologies to find the best tree (large parsimony problem). One of the drawbacks of this approach is that, as we have seen, the tree space quickly becomes prohibitive. This limitation was partially addressed by Moret and coworkers [44] who develop GRAPPA which improves on BPAnalysis by computing tight bounds and efficiently pruning the tree space. Another program to reconstruct phylogenies based on the breakpoint distance is GOTREE (see Table 4). A special feature of this last tool is that it is not restricted to genomes with equal gene content.

Siepel and Moret [72] studied a different problem: the median problem for the reversal distance. They derived a branch-and-bound algorithm to prune the search space using simple geometric properties of the problem and the linear-time machinery to compute the reversal distance [1]. Concurrently, Bourque and Pevzner [15] developed a method called MGR for both the median and the full phylogeny problem that made use of properties of additive or nearly additive trees. This algorithm, combined with GRIMM [76], is applicable to unichromosomal genomes for the reversal distance and to multichromosomal genomes for a rearrangement distance that allows reversals, translocations, fusions and fissions. The main idea of the algorithm is to look for rearrangements in the starting genomes that reduce the total distance to the other genomes and iteratively “reverse history". The key is to use good criterion to chose the order in which the rearrangements are selected.

The first method that used the conservation distance as the criterion to be minimized in the phylogenetic reconstruction problem was presented by Bergeron and coworkers [6]. Even though the problem was restricted to finding an assignment of internal nodes on a fixed phylogeny (small parsimony problem), this is a promising and active area of research.

4.3 Maximum Likelihood Methods

If we make assumptions about the mechanisms of evolution and the rates at which these changes occur, we can seek the tree which is the most likely to have generated the data observed. Such methods are called maximum likelihood methods. They tend to be computationally intensive but they have the advantage of providing a global picture of the solution space in contrast to maximum parsimony which provides a unique solution for instance.

In the context of the comparative analysis of gene order, a maximum likelihood approach turns out to be quite challenging because of our incomplete understanding of the frequency of rearrangement events but mostly because of the significantly large number of potential states at internal nodes and of phylogenetic trees [70]. Nevertheless, Dicks [21] developed one such method for gene order data, but the method presented was restricted to small instances of the problem. Other promising approaches involve the construction of a Bayesian framework and the use of Markov chain Monte Carlo to sample parameter space for two unichromosomal genomes [42, 87] or m unichromosomal genomes [38, 39]. Specifically, Larget and coworkers [39] developed the program BADGER and used it to quantify the uncertainty among the relationships of metazoan phyla on the basis of mitochondrial gene orders. So far, although these frameworks are propitious, their range of applications has been limited. It will be interesting to see if these approaches can be further applied and adapted to larger and also multichromosomal genomes.

5 Recent Applications

We have already seen some applications in which genome rearrangements acted as complementary phylogenetic characters to study evolutionary relationships in a group of organisms such as mitochondria, chloroplasts, viruses or small regions of larger genomes [3, 12, 20, 27, 51, 53, 54, 63]. We will now show how the same concepts and methodologies can be applied to compare entire eukaryotic genomes. Apart from the topology of the phylogeny, interesting questions arise from studying rates of rearrangements, types of rearrangements and predictions at ancestral nodes. We will also present some preliminary work studying genome rearrangements in cancer.

5.1 Rearrangements in Large Genomes

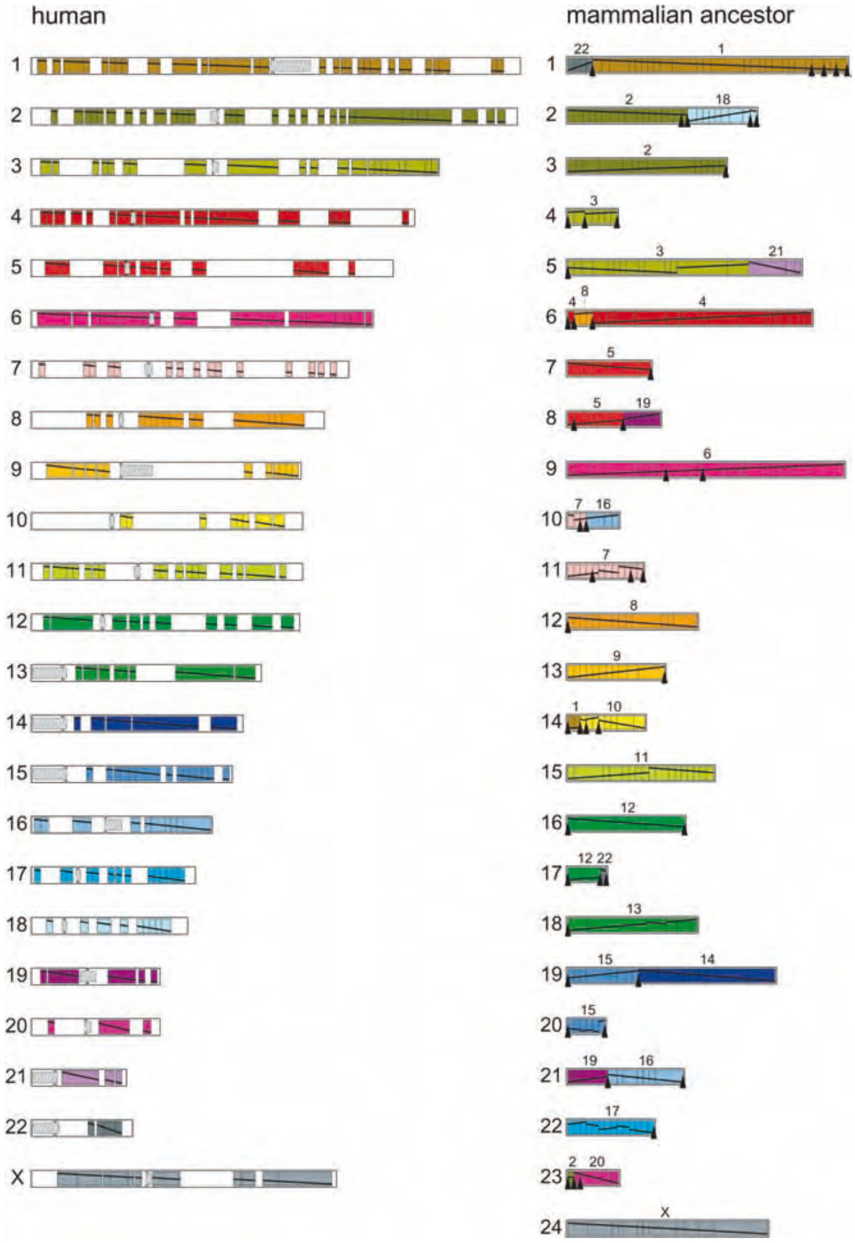
Genome rearrangements studies have traditionally been based on the relative order of homologous genes; however, as hinted at in Section 2.1, they can also be based on the relative order of a common set of homology synteny blocks (HSBs). These blocks can be defined either directly from sequence similarity [36, 56] or from the clustering of homologous genes [88]. In this context, rearrangement studies for large genomes will be reconfigured into a two-step process:

(i) Identification of HSBs shared by the set of genomes under study.

(ii) Genome rearrangement analysis of the HSBs.

In Step (i), both for sequence-based and gene-based HSBs, thresholds need to be set to allow the HSBs to extend over minor local inconsistencies that could stem from different sources: sequencing and assembly errors, small rearrangement events not enclosed in the rearrangement model of Step (ii) (e.g. transposons), inaccurate prediction of orthologous genes (e.g. in the presence of many paralogous copies), etc. For the identification of HSBs, there are advantages to using both sequence and gene data.

The most important benefit of using raw sequence data is probably to circumvent the limitation of analyzing strictly coding regions (these regions only cover a small portion of the eukaryotic genomes). Other benefits include that it avoids annotation problems, it is less sensitive to gene families and, finally, it preserves additional information on *micro-rearrangements* (rearrangements within HSBs) that can then be used as additional independent phylogenetic characters [13]. Advantages of using gene-based HSBs are that it focuses


human
mammalian ancestor
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
X
1 22 1
2 2 18
3 2
4 3
5 3 21
6 4 8 4
7 5
8 5 19
9 6
10 7 16
11 7
12 8
13 9
14 1 10
15 11
16 12
17 12 22
18 13
19 15 14
20 15
21 19 16
22 17
23 2 20
24 X

the investigations on critical regions of the genome, the thresholds are length independent and it avoids some of the noise created by repeat regions.

After Step (i), the comparison of the respective arrangements of the HSBs in the different genomes can be performed using the models, algorithms and programs described in Sections 3 and 4. This two-step analysis was used to compare the human with the mouse genome [56] and suggested a larger number inversions than previously expected [48]. It also helped motivate a model for chromosome evolution in which some breakpoints are reused nonrandomly [40, 57].

When many genomes are compared, rearrangement analysis provides information not only on phylogenetic relationships, but also on rates of rearrangements and on putative genomic architecture of ancestral genomes [13, 14, 46, 47]. For instance, the availability of the rat genome [26] allowed a comparative study with the human and the mouse [14] that confirmed an observation made using lower-resolution studies that rodent genomes have had an accelerated rate of inter-chromosomal rearrangements (e.g. translocations, fusions and fissions). The same study also conjectured on the genomic architecture of the putative murid rodent ancestor. The addition of the chicken genome [32] acting as an outgroup allowed us to look further back in time and predict the potential architecture of the mammalian ancestor [13]. This analysis also suggested:

- Variable rates of inter-chromosomal rearrangements across lineages.
- High ratio of intra-chromosomal versus inter-chromosomal rearrangements in the chicken lineage.
- Low rate of rearrangements in chicken, in the early mammalian ancestor or in both.

More recently, a comprehensive analysis of eight mammalian genomes, three sequenced genomes (human–mouse–rat) and five with high-resolution

←

Figure 4 Inferred genomic architecture of the mammalian ancestor (adapted from Ref. [46]). Each human chromosome is assigned a unique color and is divided into HSBs. These HSBs correspond to stretches of DNA for which sufficient similarity has been retain to unambiguously allow the identification of the homologous regions in all other species. The size of each block is approximately proportional to the actual size of the block in human. In human, blocks are arranged on each chromosome from left (p-arm) to right (q-arm) and physical gaps between blocks are shown to give an indication of coverage. Numbers above the rec onstructed ancestral chromosomes indicate the human chromosome homolog. Diagonal lines within each block indicate their relative order and orientation. Black arrows under the ancestral chromosome indicate that the two adjacent HSBs separated by the arrow were not found in every one of the most parsimonious solutions explored; these are considered *weak adjacencies*.

radiation-hybrid maps (cat–dog–cow–pig–horse), afforded a detailed analysis of the dynamics of mammalian chromosome evolution [46]. This study also produced a refined model of the genomic architecture of the mammalian ancestor, see Figure 4.

Applications focusing on specific areas of the genomes allow for the identification of very detailed scenarios. For instance, in the results of the study by Murphy and coworkers [46], it is possible to focus exclusively on the HSBs found on human chromosome 17; there are 14 such blocks. Chromosome 17 is interesting because, similarly to the X chromosome, it has seldom exchanged genetic material with other chromosomes during mammalian evolution. Specifically, the 14 blocks are found in one contiguous segment on a single chromosome in mouse, rat, cat and pig. They are found in two contiguous pieces on two chromosomes in cow and in three contiguous pieces on three chromosomes in dog (horse is left out of this analysis because of insufficient data). See Figure 5 for a parsimonious rearrangement scenario describing the mammalian history of this chromosome. This example once again seems to point towards uneven rates of rearrangements with no rearrangement between the cetartiodactyl ancestor and pig, but five rearrangements in cow during the same period of evolution. According to this reconstruction, the pig chromosome 12 (the homolog of human chromosome 17) is ancestral in the sense that no large-scale rearrangement has occurred on it since the divergence of these species.

5.2 Genomes Rearrrangements and Cancer

The previous section described examples of the use of genome rearrangements to study the evolution of a group of organisms. Now, because a rapid increase of chromosomal mutations is frequently observed in cancer cells, it is possible to study the cancer genome very much like as it was a new organism that had recently diverged from the normal human genome. The interest is that although cancer progression is frequently associated with genome rearrangements, the mechanisms behind these rearrangements are still poorly understood. There are many challenges in studying rearrangements in cancer cells: the heterogeneity of the cells, the complexity of the rearrangements (which include translocations, but also frequent duplications), but mostly the fact that detailed sequence is only sparsely available. So far, the cost of sequencing has been a prohibitive factor preventing large cancer genome sequencing projects, but new emerging sequencing techniques such as End Sequence Profiling [80] and Ditags [49] might help alleviate this problem. Such techniques justify the development of algorithms and tools, related to the analysis of genome rearrangement, to extract detailed tumor architecture from such data sets [59, 60].

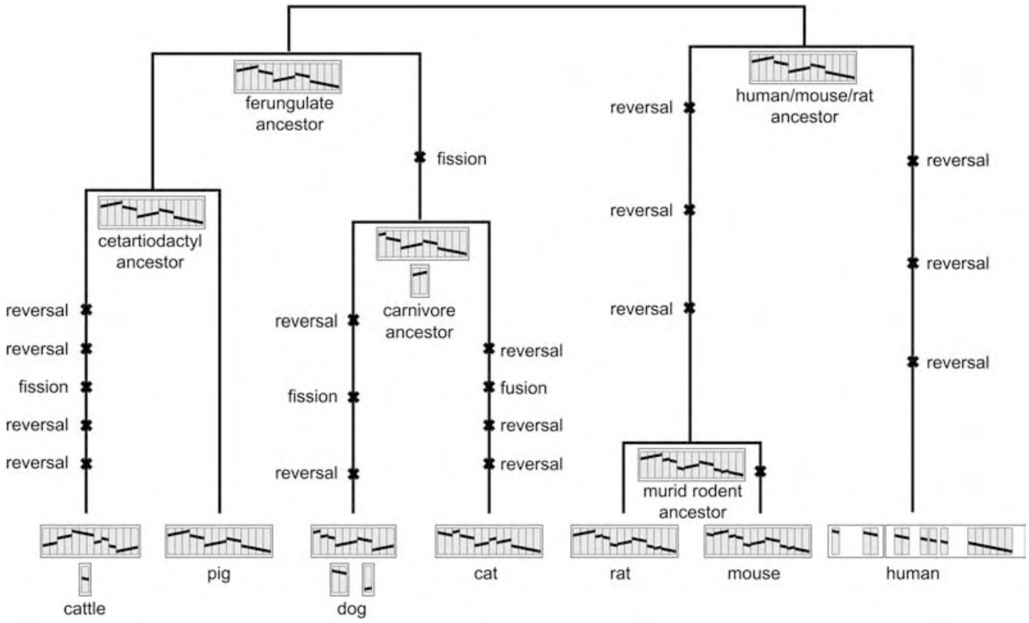


Figure 5 Mammalian history of human chromosome 17. The arrangements of 14 blocks (stretches of DNA) from human chromosome 17 with syntenic counterparts in seven other mammalian genomes (mouse, rat, cat, dog, pig and cattle) are shown at the bottom of the tree. Blocks are drawn proportionally to their size in human. A diagonal line traverses the blocks to show their order and relative orientation. In human, blocks are arranged from left (p-arm) to right (q-arm) and physical gaps between blocks are shown to give an indication of coverage. In other species, the same blocks are drawn also from left to right but in some cases these blocks are found on multiple chromosomes [cattle (2), dog (3) and carnivore ancestor (2)]. Crosses on the edges of the tree are labeled and indicate putative rearrangement events even though their exact timing is unknown. Data adapted from Ref. [46].

A complementary approach for the study of rearrangements in cancer involves looking at breakpoint regions. Many such regions have already been characterized in a large population of cancer patients [43]. Studying their distribution with respect to either chromosomal location [65] or evolutionary breakpoints [46] (identified from multispecies comparisons) is likely to provide invaluable information on the forces acting on these aberrant genomes.

6 Conclusion

6.1 Challenges

Comparative analyses of gene order would greatly benefit from established benchmarking data sets. These instances could be use to compare and refine current approaches for the study of genome rearrangements. Of course, the challenge is that for data sets generated from real genomes, the actual rearrangement history for these organisms is unknown. Thus, recovered scenarios can only be evaluated with respect to some limited aspects of their solution such as topology of the recovered tree [12, 20]. This is a suitable criterion to evaluate the merits of an approach because the topology can also be inferred from alternative, more traditional, approaches such as the comparison of individual genes. Even then, ambiguities will remain since for many interesting sets of species, some aspects of the topology are debatable (e.g. especially when deep branches are involved) and the information extracted from genome rearrangements might be different from that provided from sequence analysis but not necessarily erroneous.

Other criteria that can be used for the evaluation of solutions are some of the coarse features of the recovered ancestors such as the ancestral chromosomal associations. These are associations between modern chromosomes (e.g. human chromsomome) that are inferred to have been present in the ancestors [73]. Unfortunately, once again, definitive evaluation is difficult for two reasons: (i) the expected associations rely on low-density comparative maps and are likely to be incomplete, and (ii) multiple alternative ancestors are typically recovered in rearrangement studies making more than a single prediction [13, 14]. Now that high-quality sequences are increasingly becoming available for many genomes, one would actually expect to see the knowledge on such associations to be expanded and refined, especially after carrying out combinatorial analyses that take into account more than just co-occurrences.

A logical alternative to real benchmarking data sets with unknown rearrangement history is provided by simulated data sets. Unfortunately, there are drawbacks inherent to this approach as well. In particular, simulated data sets will always bias the evaluation towards approaches that have an underlying rearrangement model that is most compatible with the model that was used to generate the data. Such data sets can be a great asset in evaluating alternative methods that have the same assumptions, but they are of limited value in identifying whether a particular method will be successful on real data.

Another desired development would be a more systematic study and comparison of different distance criterion. Specifically, with the development of new measures [7,31,78], a detailed analysis of the strengths and weaknesses of the different approaches is needed to assess the context in which they are most applicable. For instance, model-free measures such as the breakpoint distance and the conservation distance are probably the most appropriate when the underlying rearrangements follow uncharacterized rules.

Finally, a key challenge associated with this type of analysis involves studying the causes and consequences of genome rearrangements. Although these events are well characterize in both evolution and cancer, the extent of the biological repercussions is still unclear. For instance, large rearrangement events can have a significant impact at the population level by creating subpopulations for which recombination in the affected region will be impossible but the question of whether such events also play a role in speciation for instance is still debated. On a different topic, is there a faster phenotypic evolution associated with a faster rearrangement rate? Given the amount of comparative data recently made available [26, 32, 37, 79, 84], the hope is that some answers might be within reach.

In order to start exploring these questions, looking at sets of highly diverse genomes spanning long evolutionary distances is not the most appropriate. Inherent to such data sets will always be ambiguities such as the accuracy of the rearrangement model, the quality of the solution obtained, the order of rearrangements found on edges of the phylogeny, the presence of alternative ancestors and the presence of alternative rearrangement scenarios. A more practical framework in which to ask questions about the impact of genome rearrangements would probably involve looking at more closely related species where the inferred rearrangement scenario is less disputable.

6.2 Promising New Approaches

The rearrangement model will always have a critical impact on the reconstructed scenario. In many of the applications presented [13, 14, 46], the rearrangement model includes reversals, translocations, fusions and fissions, but these events are considered equally likely (i.e. the weight of each of the events is the same when the distance is computed). In reality, short reversals are probably more common than fusions for instance. Consider the carnivore ancestor shown Figure 5; in the displayed solution, there is a fission between the ferungulate ancestor and the carnivore ancestor followed by a fusion in the cat lineage. An alternative solution exists with the same total number of rearrangements, but in which this fission plus fusion is replaced by a single fission on the dog lineage and a reversal in the cat lineage. Such a scenario is probably more realistic than the one displayed but it is masked by our assumption of equally likely events. Perhaps approaches with weighted events, such as in Blanchette and coworkers [11], or approaches that make use of a maximum likelihood framework, such as Larget and coworkers [39], could help alleviate some of these ambiguities.

Although strictly incorporating transpositions into the rearrangement model remains computationally challenging, there is renewed interest in allowing block interchanges, an operation which includes all types of transposition [19, 41, 86]. The inclusion of this process actually allows a dramatic simplification of the HP theory (see Section 3.2) [86] and is likely to enable new applications.

Nevertheless, because rearrangement models are always debatable, model-free approaches that make use of breakpoint or conservation distance, such as Bergeron and coworkers [6], are also attractive and interesting. Hopefully these approaches will be extended and applied to a larger variety of problems.

Finally, another promising area of research is the analysis of breakpoint regions. These regions typically contain an unusual mosaic of content [36,77] and they are also likely to harbor information on the mechanisms behind the rearrangements that created them. In the context of cancer, these are also the regions that have the potential to host the destructive fusion genes. Comparing cancer breakpoints with evolutionary breakpoints [46] might provide some information on the forces shaping the genomic architecture of modern organisms.

Acknowledgments

The author is supported by funding from the Biomedical Research Council (BMRC) of the Agency for Science, Technology and Research (A*STAR) in Singapore.

References

1 Bader, D., B. Moret and M. Yan. 2001. A linear-time algorithm for computing inversion distance between signed permutations with an experimental study. In Proc. 7th Int. Workshop on Algorithms and Data Structures, Providence, RI, USA: 365–76.

2 Bafna, V. and P. A. Pevzner. 1998. Sorting by transpositions. SIAM J. on Discrete Mathematics **11**: 224–40.

3 Bafna, V. and P. Pevzner. 1995. Sorting by reversals: genome rearrangements in plant organelles and evolutionary history of X chromosome. Mol. Biol. Evol. **12**: 239–46.

4 Bafna, V. and P. Pevzner. 1996. Genome rearrangements and sorting by reversal. SIAM J. Comput. **25**: 272–89.

5 Bergeron, A., J. Mixtacki and J. Stoye. 2004. Reversal distance without hurdles and fortresses. In Proc. Annu. Symp. on Combinatorial Pattern Matching, Istanbul, Turkey: 388–99.

6 Bergeron, A., M. Blanchette, A. Chateau and C. Chauve. 2004. Reconstructing ancestral gene orders using conserved intervals. in Proc. WABI, Bergen, Norway: 14–25.

7 Bergeron, A. and J. Stoye. 2003. On the similarity of sets of permutations and its applications to genome comparison. In Proc. COCOON, Big Sky, MT, USA: 68–79.

8 Bergeron, A. 2001. A very elementary presentation of the Hannenhalli–Pevzner theory. In Proc. Annu. Symp. on Combinatorial Pattern Matching, Jerusalem, Israel: 106–17.

9 Berman, P. and S. Hannenhalli. 1996. Fast sorting by reversal. In Proc. Annu. Symp. on Combinatorial Pattern Matching, Laguna Beach, CA, USA: 168–85.

10 Blanchette, M., G. Bourque and D. Sankoff. 1997. Breakpoint phylogenies. In Proc. Genome Informatics Workshop, Tokyo, Japan: 25–34.

11 Blanchette, M., T. Kunisawa and D. Sankoff. 1996. Parametric genome rearrangement. Gene **172**: GC11–7.

12 BLANCHETTE, M., T. KUNISAWA AND D. SANKOFF. 1999. Gene order breakpoint evidence in animal mitochondrial phylogeny. J. Mol. Evol. **49**: 193–203.

13 BOURQUE, G., E. ZDOBNOV, P. BORK, P. PEVZNER AND G. TESLER. 2005. Comparative architectures of mammalian and chicken genomes reveal highly variable rates of genomic rearrangements across different lineages. Genome Res. **15**: 98–110.

14 BOURQUE, G., P. PEVZNER AND G. TESLER. 2004. Reconstructing the genomic architecture of ancestral mammals: lessons from human, mouse, and rat genomes. Genome Res. **14**: 507–16.

15 BOURQUE, G. AND P. PEVZNER. 2002. Genome-scale evolution: reconstructing gene orders in the ancestral species. Genome Res. **12**: 26–36.

16 BRYANT, D. 1998. The complexity of breakpoint median problem. *Technical Report CRM-2579*. Centre de recherches mathématiques, Université de Montréal.

17 CAPRARA, A. 1997. Sorting by reversals is difficult. Proc. RECOMB **1**: 75–83.

18 CAPRARA, A. 1999. Formulations and complexity of multiple sorting by reversals. Proc. RECOMB **3**: 84–93.

19 CHRISTIE, D. A. 1996. Sorting permutations by block-interchanges. Inf. Process. Lett. **60**: 165–9.

20 COSNER, M., R. JANSEN, B. MORET, L. RAUBESON, L. WANG, T. WARNOW AND S. WYMAN. 2002. A new fast heuristic for computing the breakpoint phylogeny and experimental phylogenetic analyses of real and synthetic data. Proc. ISMB **8**: 104–15.

21 DICKS, J. 2002. CHROMTREE: maximum likelihood estimation of chromosomal phylogenies. In SANKOFF D. AND J. H. NADEAU (eds.), *Comparitive Genomics (DCAF-2000).* Kluwer, Dordrecht: 333–42. 2000.

22 DOBZHANSKY, T. AND A. STURTEVANT. 1938. Inversions in the chromosomes of *Drosophila pseudoobscura*. Genetics **23**: 28–64.

23 EARNEST-DEYOUNG, J., E. LERAT AND B. MORET. 2004. Reversing gene erosion – reconstructing ancestral bacterial genomes from gene-content and order data. In Proc. WABI, Bergen, Norway: 1–13.

24 EL'MABROUK, N. 2005. Genome rearrangement with gene families. In GASCUEL O. (ed.), *Mathematics of Evolution and Phylogeny*. Oxford University Press, Oxford: 291–313.

25 FITCH, W. 1977. On the problem of discovering the most parsimonious tree. Amer. Natur. **111**: 223–57.

26 GIBBS, R., G. WEINSTOCK, M. METZKER, ET AL. 2004. Genome sequence of the Brown Norway rat yields insights into mammalian evolution. Nature **428**: 493–521.

27 HANNENHALLI, S., C. CHAPPEY, E. KOONIN AND P. PEVZNER. 1995. Genome sequence comparison and scenarios for gene rearrangements: a test case. Genomics **30**: 299–311.

28 HANNENHALLI, S. AND P. PEVZNER. 1995. Transforming cabbage into turnip (polynomial algorithm for sorting signed permutations by reversals). Proc. 27th Annu. ACM–SIAM Symp. on the Theory of Computing, Las Vegas, NV, USA: 178–89.

29 HANNENHALLI, S. AND P. PEVZNER. 1995. Transforming men into mice: polynomial algorithm for genomic distance problem. Proc. 36th IEEE Symp. on Foundations of Computer Science, Los Alamitos, CA, USA: 581–92.

30 HANNENHALLI, S. AND P. PEVZNER. 1999. Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. J. ACM **46**: 1–27.

31 HEBER, S. AND J. STOYE. 2001. Finding all common intervals of *k* permutations. In Proc. Annu. Symp. on Combinatorial Pattern Matching, Jerusalem, Israel: 207–18.

32 HILLIER, L., W. MILLER, E. BIRNEY, ET AL. 2004. Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. Nature **432**: 695–716.

33 JARNIK, V. 1934. Sur les graphes minima, contenant n points donnés. Cas. Pest. Mat. **63**: 223–35.

34 KAPLAN, H., R. SHAMIR AND R. TARJAN. 1997. Faster and simpler algorithm for sorting signed permutations by reversals. Proc. 8th Annu. ACM–SIAM Symp. on Discrete Algorithms, New Orleans, LA, USA: 344–51.

35 KECECIOGLU, J. AND D. SANKOFF. 1994. Efficient Bounds for oriented chromosome inversion distance. In Proc. Annu. Symp. on Combinatorial Pattern Matching, Asilomar, CA, USA: 307–25.

36 KENT, W., R. BAERTSCH, A. HINRICHS, W. MILLER AND D. HAUSSLER. 2003. Evolution's cauldron: duplication, deletion, and rearrangement in the mouse and human genomes. Proc. Natl Acad. Sci. USA **100**: 11484–9.

37 LANDER, E. S., L. M. LINTON, B. BIRREN, ET AL. 2001. Initial sequencing and analysis of the human genome. Nature **409**: 860–921.

38 LARGET, B., D. SIMON AND J. KADANE. 2002. Bayesian phylogenetic inference from animal mitochondrial genome arrangements (with discussion). J. R. Stat. Soc. **B**: 681–93.

39 LARGET, B., D. SIMON, J. KADANE AND D. SWEET. 2005. A Bayesian analysis of metazoan mitochondrial genome arrangements. Mol. Biol. Evol. **22**: 486–95.

40 LARKIN, D., A. E. VAN DER WIND, M. REBEIZ ET AL. 2003. A cattle–human comparative map built with cattle BAC-ends and human genome sequence. Genome Res. **13**: 1966–72.

41 LIN, Y. C., C. L. LU, H. Y. CHANG AND C. Y. TANG. 2005. An efficient algorithm for sorting by block-interchanges and its application to the evolution of *Vibrio* species. J. Comput. Biol. **12**: 102–12.

42 MIKLOS, I. 2003. MCMC genome rearrangement. Bioinformatics **19**: ii130–7.

43 MITELMAN, F., B. JOHANSSON AND F. MERTENS. 2005. Mitelman Database of Chromosome Aberrations in Cancer. http://cgap.nci.nih.gov/Chromosomes/Mitelman.

44 MORET, B., S. WYMAN, D. BADER, T. WARNOW AND M. YAN. 2001. A new implementation and detailed study of breakpoint analysis. Pac. Symp. Biocomput. **6**: 583–94. 2001.

45 MORGAN, T. AND C. BRIDGES. 1916. Sex-linked inheritance in *Drosophila*. Carnegie Inst. Washington Publ. **237**: 1–88.

46 MURPHY, W., D. LARKIN, A. E. VAN DER WIND, G. BOURQUE, ET AL. 2005. Dynamics of mammalian chromosome evolution inferred from multispecies comparative maps. Science **309**: 613–17.

47 MURPHY, W., G. BOURQUE, G. TESLER, P. PEVZNER AND S. O'BRIEN. 2003. Reconstructing the genomic architecture of mammalian ancestors using multispecies comparative maps. Hum. Genomics **1**: 30–40.

48 NADEAU, J. AND B. TAYLOR. 1984. Lengths of chromosomal segments conserved since divergence of man and mouse. Proc. Natl Acad. Sci. USA **81**: 814–8.

49 NG, P., C. WEI, W. SUNG ET AL. 2005. Gene identification signature (GIS) analysis for transcriptome characterization and genome annotation. Nat. Methods **2**: 105–11.

50 O'BRIEN, S., M. MENOTTI-RAYMOND, W. MURPHY, ET AL. 1999. The promise of comparative genomics in mammals. Science **286**: 458–81.

51 OLMSTEAD, R. AND J. PALMER. 1994. Chloroplast DNA systematics: a review of methods and data analysis. Am. J. Bot. **81**: 1205–24.

52 OZERY-FLATO, M. AND R. SHAMIR. 2003. Two notes on genome rearrangement. J. Bioinf. Comput. Biol. **1**: 71–94.

53 PALMER, J. AND L. HERBON. 1988. Plant mitochondrial DNA evolves rapidly in structure, but slowly in sequence. J. Mol. Evol. **27**: 87–97.

54 PALMER, J. 1992. Chloroplast and mitochondrial genome evolution in land plants. In HERRMANN, R. (ed.), *Cell Organelles.* Springer, Berlin: 99–133.

55 PE'ER, I. AND R. SHAMIR. 1988. The median problem for breakpoints are NP-complete. Electronic Colloqium on Computational Complexity. Technical Report **TR98-071**.

56 Pevzner, P. and G. Tesler. 2003. Genome rearrangements in mammalian evolution: lessons from human and mouse genomes. Genome Res. **13**: 37–45.

57 Pevzner, P. and G. Tesler. 2003. Human and mouse genomic sequences reveal extensive breakpoint reuse in mammalian evolution. Proc. Natl Acad. Sci. USA **100**: 7672–7.

58 Pevzner, P. 2000. *Computational Molecular Biology: An Algorithmic Approach.* MIT Press, Cambridge, MA.

59 Raphael, B., S. Volik, C. Collins and P. Pevzner. 2003. Reconstructing tumor genome architectures. Bioinformatics **19**: ii162–71.

60 Raphael, B. and P. Pevzner. 2004. Reconstructing tumor amplisomes. Bioinformatics **20 (Suppl 1)**: i265–73.

61 Rokas, A. and P. Holland. 2000. Rare genomic changes as a tool for phylogenetics. Trends Ecol. Evol. **15**: 454–9.

62 Saitou, N. and M. Nei. 1987. The neighbor-joining method: A new method for reconstructing phylogenetic trees. Mol. Biol. Evol. **4**: 406–25.

63 Sankoff, D., G. Leduc, N. Antoine, B. Paquin, B. Lang and R. Cedergren. 1992. Gene order comparisons for phylogenetic inference: evolution of the mitochondrial genome. Proc. Natl Acad. Sci. USA **89**: 6575–9.

64 Sankoff, D., G. Sundaram and J. Kececioglu. 1996. Steiner points in the space of genome rearrangements. Int. J. Found. of Comp. Sci. **7**: 1–9.

65 Sankoff, D., M. Deneault, P. Turbis and C. Allen. 2002. Chromosomal distributions of breakpoints in cancer, infertility, and evolution. Theor. Popul. Biol. **61**: 497–501.

66 Sankoff, D. and J. H. Nadeau (eds). 2000. *Comparative Genomics: Gene Order Dynamics, Comparative Maps and Multigene Families.* Kluwer, Dordrecht.

67 Sankoff, D. and M. Blanchette. 1997. The median problem for breakpoints in comparative genomics. In Proc. COCOON, Shanghai, China: 251–63.

68 Sankoff, D. 1992. Edit distance for genome comparison based on non-local operations. In: Proc. Annu. Symp. on Combinatorial Pattern, Tucson, AZ, USA: 121–35.

69 Sankoff, D. 1999. Genome rearrangement with gene families. Bioinformatics **15**: 909–17.

70 Savva, G., J. Dicks and I. Roberts. 2003. Current approaches to whole genome phylogenetic analysis. Brief. Bioinformat. **4**: 63–74.

71 Setubal, J. and J. Meidanis. 1997. *Introduction to Computational Molecular Biology.* PWS Publishing, Boston, MA.

72 Siepel, A. and B. Moret. 2001. Finding an optimal inversion median: experimental results. In Proc. WABI, Aarhus, Denmark: 189–203.

73 Stanyon, R., G. Stone, M. Garcia and L. Froenicke. 2003. Reciprocal chromosome painting shows that squirrels, unlike murid rodents, have a highly conserved genome organization. Genomics **82**: 245–9.

74 Sturtevant, A. H. 1921. Genetic studies on *Drosophila simulans*. II. Sex-linked group of genes. Genetics **6**: 43–64.

75 Tesler, G. 2002. Efficient algorithms for multichromosomal genome rearrangements. J. Comput. Syst. Sci. **65**: 587–609.

76 Tesler, G. 2002. GRIMM: genome rearrangements web server. Bioinformatics **18**: 492–3.

77 Trinh, P., A. McLysaght and D. Sankoff. 2004. Genomic features in the breakpoint regions between syntenic blocks. Bioinformatics **20 (Suppl. 1)**: I318–25.

78 Uno, T. and M. Yagiura. 2000. Fast algorithms to enumerate all common intervals of two permutations. Algorithmica **26**: 290–309.

79 Venter, J., M. Adams, E. Myers, et al. 2001. The sequence of the human genome. Science **291**: 1304–51.

80 Volik, S., S. Zhao, K. Chin, J. Brebner, et al. 2003. End-sequence profiling: sequence-based analysis of

aberrant genomes. Proc. Natl Acad. Sci. USA **100**: 7696–701.

81 WALTER, M., Z. DIAS AND J. MEIDANIS. 2000. A new approach for approximating the transposition distance. In Proc. Seventh International Symposium on String Processing Information Retrieval, La Coruna, Spain: 199–208.

82 WANG, L., R. JANSEN, B. MORET, L. RAUBESON AND T. WARNOW. 2002. Fast phylogenetic methods for the analysis of genome rearrangement data: an empirical study. Pac. Symp. Biocomput. **7**: 524–35.

83 WANG, L.-S. AND T. WARNOW. 2001. Estimating true evolutionary distances between genomes. Proc. 33rd Symp. on Theory of Computing, Heraklion, Crete, Greece: 637–46.

84 WATERSTON, R., K. LINDBLAD-TOH, E. BIRNEY, ET AL. 2002. Initial sequencing and comparative analysis of the mouse genome. Nature **420**: 520–62.

85 WATTERSON, G., W. EWENS, T. HALL AND A. MORGAN. 1982. The chromosome inversion problem. J. Theor. Biol. **99**: 1–7.

86 YANCOPOULOS, S., O. ATTIE AND R. FRIEDBERG. 2005. Efficient sorting of genomic permutations by translocation, inversion and block interchange. Bioinformatics **21**: 3340–6.

87 YORK, T., R. DURRETT AND R. NIELSEN. 2002. Bayesian estimation of the number of inversions in the history of two chromosomes. J. Comput. Biol. **9**: 805–18.

88 ZDOBNOV, E., C. VON MERING, I. LETUNIC, ET AL. 2002. Comparative genome and proteome analysis of *Anopheles gambiae* and *Drosophila melanogaster*. Science **298**: 149-59.

Part 4 Molecular Structure Prediction

9
Predicting Simplified Features of Protein Structure

Dariusz Przybylski and Burkhard Rost

1 Introduction

1.1 Protein Structures are Determined Much Slower than Sequences

At the end of 2005 there were about 30 000 experimentally determined protein three-dimensional (3-D) structures in public databases [17]. At the same time there were almost 40 million genes known [16] and approximately 1.5 million verified [11] protein sequences. This gap between structure and sequence continues to grow – despite successful efforts at large-scale structure determination ("structural genomics" [118, 150]), the rate of new structures (thousands per year) continues to increase much slower than the rate of new sequences (many millions per year). Moreover, experimental structure determination has been largely or entirely unsuccessful for important classes such as cell membrane proteins.

1.2 Reliable and Comprehensive Computations of 3-D Structures are not yet Possible

In principle, we could compute 3-D structures from sequences using basic physical principles [9]. However, the complexity of the problem exceeds by far today's computational resources. Speeding up molecular dynamics by a factor of 1000 appears an objective within reach to Schroedinger Inc. While this would undoubtedly yield important insights into the problem, it may still not bring reliable predictions of 3-D structures from sequence. Even given infinite CPU resources, another serious obstacle is raised by the

Bioinformatics – From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

minute energy differences between native and unfolded structures (around 1 kcal mol^{-1}). This minute difference along with the uncertainty in estimating constants needed for calculations based on first principles makes it very difficult to find an approximate approach that is both simple and sufficiently accurate. Although we cannot model from sequence, comparative modeling yields rather accurate predictions based on sequence homology to proteins of known structure [101]. Such modeling is based on the fact that proteins with similar sequences usually have similar structures. Assume we know the structure for K and that we want to predict the structure for U that is sequence-similar to K. Comparative modeling simply predicts U to have the same structure as K and models the structure of U based on the known backbone of K. However, for the majority of protein sequences no sufficiently detailed structural information is available or computable.

1.3 Predictions of Simplified Aspects of 3-D Structure are often very Successful

In the absence of experimental or predicted 3-D structures, many researchers concentrate on trying to simplify the problem and predict particular structural features. One of the first well-defined problems was the prediction of protein secondary structure. Progress in this field has been steady and current secondary structure predictions are useful for many biological applications. Techniques that were developed in the context of secondary structure predictions were successfully applied to the prediction of many other aspects of protein structure such as solvent accessibility, inter-residue contact maps, disordered regions, domain organization and specialized for distinctive cases such as transmembrane regions of proteins.

2 Secondary Structure Prediction

2.1 Assignment of Secondary from 3-D Structure

2.1.1 Regular Secondary Structure Formation is Mostly a Local Process

Three-dimensional structures exhibit extensive local conformational regularities known as regular secondary structure. These local structures (most importantly helices and sheets) can be described as ordered arrangements of a polypeptide chain without reference to amino acid type or actual 3-D conformations. They are stabilized primarily by hydrogen bonds formed between the atoms present in the polypeptide backbone, but interactions with solvent and other protein atoms also play an important role. It is believed that the formation of secondary structure is an important step toward folding. Identifying the rules for packing the elements of secondary structure against each other would afford the derivation of a very limited number of possible stable conformations. Unfortunately, the formation of secondary structure is not entirely a local process. Thus, a perfect prediction of secondary structure without knowledge of nonlocal information is unlikely. Note that secondary structure can be written in a string of assignments for each residue, i.e. it is essentially a 1-D feature of protein 3-D structure. (Unfortunately, some authors are lured into misusing the term 2-D structure, possibly in response to a misunderstanding of the word "secondary".)

2.1.2 Secondary Structures can be Somehow Flexible

Regular secondary structure is a striking, macroscopically visible aspect of 3-D structure. However, secondary structures are not rigid. Calculations and experiments indicate that structural shifting occurs, especially in surface regions. The adoption of a particular structure may depend on many environmental factors. This is illustrated by the fact that sometimes the secondary structure states differ among various crystals of the same protein as well as various nuclear magnetic resonance (NMR) models by as much as 5–15%. This variability constrains the upper limit of what we can expect from prediction methods – arguably levels of about 90% (percentage of residues predicted correctly in either of the three states helix, strand, other). While many residues can be confidently classified into one of the secondary structure types, there are also those for which classification is ambiguous. This problem is especially evident at terminal locations of secondary structure elements; it is just another aspect of the observation that protein structures are dynamic objects. Historically, assignments were carried out through visual inspection by experimentalists. That approach introduced a human-based inconsistency. In 1983, this inconsistency was first addressed by an objective, automatic assignment method [Dictionary of Secondary Structure of Proteins (DSSP), see below]. Many such methods followed; they all apply criteria consistent for all proteins but they often differ between each other.

2.1.3 Automatic Assignments of Secondary Structure

The first assignments of protein secondary structure were carried out by Pauling and others [126] even before experimental 3-D structures of proteins became available. They were based on intra-backbone hydrogen bonds. One of the first and most popular automatic methods, DSSP [76], used a similar approach. The DSSP method calculates the interaction energy between backbone atoms based on an electrostatic model [76]. It assigns a hydrogen bond if the interaction energy is below a chosen threshold (–0.5 kcal mol^{-1}). The structure assignments are defined such that visually appealing and unbroken structures are formed from groups of hydrogen bonds. Another popular automatic assignment method, the STRuctural IDEntification method (STRIDE [51]) uses ϕ–ψ torsion angles and empirically derived hydrogen bond energy. The parameters used by this method are optimized to reproduce visual assignments provided by experimentalists determining 3-D structures and so in effect the method averages out human bias. The method DEFINE [143] assigns secondary structure using C_α coordinates. The assignment is carried out through comparison of observed C_α distances with those derived from ideal secondary structures. If the distances are within set discrepancy limits, then the secondary structure is assigned. The method P-Curve [173] makes assignments based on geometrical analysis of protein curvature. It uses differential geometry-based representations of standard structural motifs and through a set of geometrical transformations tries to match these motifs with those found in known 3-D structures. P-Curve assignments differ significantly from those based on hydrogen bonds and/or ϕ–ψ torsion angles. DSSP, P-Curve and Define assignment methods agree for only about two-thirds of all residues [30]. There are various reasons for disagreements; the most important one may simply be that secondary structure is dynamic, i.e. that there simply is no such thing as a secondary structure "state". This problem is reflected in the DSSPcont method that introduces continuous secondary structure assignments [8]. The continuum results from calculations of weighted averages of DSSP assignments that are based on various hydrogen bond energy thresholds. As a result, each protein residue is assigned with likelihoods of all secondary structure states. Residues that have a higher probability for a single "state" appear to also be more rigid according to NMR measurements of motions on timescales important for protein function [8]. Other, more application-oriented approaches to defining local structures are possible. For example, one may try to define a new secondary structure alphabet with a goal of improving fold recognition algorithms [78]. The numerical values of prediction accuracy presented in this chapter are based on the most widely used DSSP assignment. Evaluations based on STRIDE tend to yield higher values and no state-of-the-art prediction method has been evaluated on P-Curve.

2.1.4 **Reduction to Three Secondary Structure States**

DSSP distinguishes eight different "states": three types of helical structures [α-helix ("H", four-residue period), π-helix ("I", five-residue period) and 3_{10}-helix ("G", 3-residue period)], extended β-sheet ("E"), β-bridge ("B"), turn ("T"), bend ("S") and other nonregular states (blank). Of those, α-helix and β-strand (Figure 1) comprise more than 50% of all protein residues. Some prediction methods attempt to predict all eight states. However, a widely used strategy is to map the eight "states" into three major "classes": helical, extended and other (often imprecisely referred to as "nonregular", "coil"

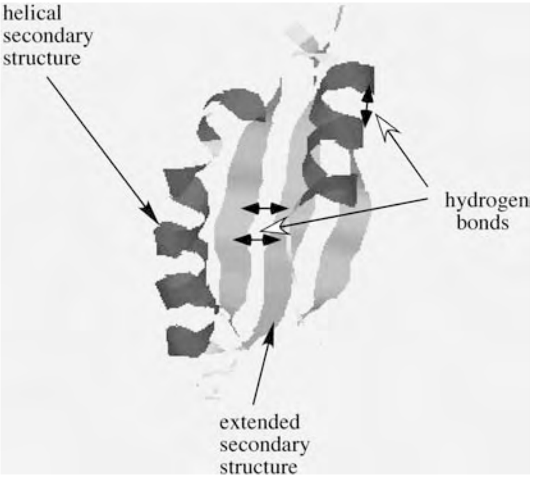


Figure 1 Ribbon diagram of protein secondary structure. Secondary structures are local arrangements of the protein backbone without reference to the amino acid type or the 3-D conformation. They are stabilized by hydrogen bonds between atoms of the main chain (backbone). Very roughly, secondary structures can be classified into three classes: helical (H), extended (E) (strand) and loopy (other) L. The figure contains a schematic representation of the E2 DNA-binding domain [21] (Protein Data Bank [17] code 1a7g).

or "turn"). Different maps are possible, but the most popular one (which incidentally is most difficult to predict [35]) is the following: [GHI] = helical ("h"), [EB] = extended ("e") and [TS] = nonregular (coil) ("I"). The alternative translation that results in seemingly higher prediction accuracies, i.e. [H] = helical, [E] = extended and [GITS] = nonregular, is sometimes used.

2.2 Measuring Performance

2.2.1 Performance has Many Aspects Relating to Many Different Measures

Depending on the application there are various views as to what constitutes a high-quality prediction. On the one hand, it is important to correctly predict the secondary structure "state" for each residue (per-residue accuracy); on the other hand, it may be more relevant to predict the coarse-grained presence of, for example, a helix than all residues in the helix (segment-based accuracy). Accordingly, many measures have been used to assess prediction quality: simple percentages of per-residue accuracy (Eq. 1), Matthew's correlation coefficients, percentage of confusion between strand and helix states [38] (Eq. 2); simple segment-based measures such as the number of correctly predicted segments, the average ratio of predicted to observed segment lengths, the difference between the distribution of predicted and observed segment lengths [156]; or the more elaborated and widely used segment overlap score *SOV* [160, 187] (Eq. 3). These are only some of the measures that have been applied. In this chapter, we focus on two measures for per-residue accuracy, i.e. percentages Q_K (Eq. 1) and the *BAD* score (Eq. 2), and one measure for per-segment accuracy, i.e. *SOV*.

2.2.2 **Per-residue Percentage Accuracy:** Q_K

Perhaps the most intuitive and simplest measure for performance is the average percentage of correctly predicted states. For a protein composed of L residues and for K possible secondary structure states the per-residue prediction accuracy Q_K is defined as:

$$Q_K = 100 \times \sum_{i=1}^{K} C_i / L \qquad (1)$$

where C_i is the number of residues correctly predicted in secondary structure state i. For a three-state alphabet this translates into a Q_3 measure. The average accuracy can be computed as an average per protein or an average per residue in which case the number of all residues is used for L.

2.2.3 **Per-residue Confusion between Regular Elements: *BAD***

Not all secondary structure prediction mistakes are equal. For instance, when using secondary structure predictions to model 3-D structure, confusing helix and extended (strand) is more detrimental than confusing regular with non-regular states. The percentage of such "bad" predictions constitutes the *BAD* score. If L is a total number of amino acid residues in a protein and *Bh* (*Be*) is the number of helical (strand) residues predicted in strand (helix) state, then the *BAD* score is expressed as:

$$BAD = 100 * \frac{Bh + Be}{L} \; . \qquad (2)$$

Two predictions with equal Q_3 and/or *SOV* scores can have very different *BAD* scores.

2.2.4 **Per-segment Prediction Accuracy: *SOV***

Regular secondary structure elements are built of continuous stretches of residues belonging to the same state, e.g. most helices are about 10 residues long. It can be argued that mis-predicting two residues at either end of a helix is not an important mistake (note: 2 + 2 out of 10 means 60% accuracy). In contrast, only predicting 60% of the helices in a protein is a severe problem. Such realities are reflected in segment-based measures. The most widely used is the segment overlap (*SOV*) measure [160,187]:

$$SOV = 100 \times \frac{1}{N} \sum_{i}^{K} \sum_{S(i)} \frac{minov(s_{\text{obs}}, s_{\text{pred}}) + \delta(s_{\text{obs}}, s_{\text{pred}})}{\text{maxov}(s_{\text{obs}} s_{\text{pred}})} \times len(s_{\text{obs}}) \qquad (3)$$

where K is the number of different secondary structure types; the second summation is over all overlapping secondary structure segments of observed s_{obs} and predicted s_{pred} secondary structure of the same type; *minov* is the number of positions at which segments overlap; *maxov* is the number of overlapping positions plus the number of remaining residues from each segment of the given pair; $len(s_{obs})$ is the length of a reference secondary structure segment (observed experimentally); N is the total number of overlapping segments pairs of the same type; and $\delta(s_{obs}, s_{pred})$ is the accepted variation between segments that assures ratio of 1.0 when the variations between s_{obs} and s_{pred} are minor. One can easily envision two different secondary structure predictions that have the same Q_3 and different *SOV* scores. For example, if instead of a observed long helix of length n one prediction consists of a shorter helix of length m and the second prediction comprises two short helices of combined length equal to m (other residues predicted as coil), then the Q_3 scores of both predictions are going to be the same while the *SOV* scores are going to be different.

2.3 Comparing Different Methods

2.3.1 Generic Problems

In this section we describe problems with the evaluation of prediction methods that are entirely generic, i.e. valid for all prediction methods. Although many ideas and concepts have been introduced to predict secondary structure and have then been used for other purposes, many of the mistakes in comparing methods have also been unraveled first and most clearly for the example of secondary structure predictions. Secondary structure prediction methods may be the only example of publications with claims to performance accuracy that survived more than a decade. (To put this into perspective: our section focuses on methods for which performance has, on average, been unusually well estimated; nevertheless, the only other field that we review for which *any* estimate survived 5 years was the prediction of solvent accessibility and the vast majority of publications in that field heavily overestimated performance!)

2.3.2 Numbers can often not be Compared between Two Different Publications

Prediction methods are often published with estimates of performance that are supported by cross-validation experiments. However, the terms "cross-validation" or the related term "jackknife" are by no means sufficiently well-defined to translate into "estimate ok". In fact, most publications make some serious mistakes as is demonstrated by the simple fact that very few estimates of performance have survived. One problem is the overlap between "training" and "testing" sets. It is trivial to reach very high performance by training on proteins that are very similar to those in the testing set. There are various strategies that deal with the similarity problem [67,188]. Another issue is that of using the performance of the test set to choose some parameters by, for example, reporting full cross-validation results for *N* different parameters and then concluding that the best of those *N* is the performance of the final method. Instead, performance estimates should always be based on a data set that was not used in ANY step of the development. However, even if we had two publications that both used cross-validation "correctly", we still cannot necessarily compare the numbers published by both directly. First, both have to have used the same standard of truth (here, the same assignment method, e.g. DSSP, and the same conversion of the eight DSSP states into three prediction classes). Second, they both have to have been based on identical test sets. Often, the test sets used by developers are not representative and differ from each other. Proteins vary in their structural complexity and such variation is correlated with prediction difficulty. We could argue that test sets should be frozen (and this has indeed been done in many cases). Such a set should be sufficiently large to allow proper evaluation of statistical differences among methods. Although a *sine qua non*, this freezing strategy does not suffice – data sets in biology change constantly, almost always more recent sets are more reliable and representative. Therefore, we also need evaluations based on sets that are as recent as possible. One way of merging these two demands is by carrying out two tests: one on a frozen set used by others and the other on a more recent set. As an aside, it is not necessary to use *n*-fold cross-validation experiments with the largest possible *n*. The exact value of *n* is not important as long as the test set is not misused for adjusting a method's parameters and it is representative of the entire structure space.

2.3.3 **Appropriate Comparisons of Methods Require Large, "Blind" Data Sets**

One of the solutions to the problem of comparing methods is to use a sufficiently large test set composed of proteins that were neither used nor are similar to any protein that was used for development of any method. This idea was first realized in the field of structure prediction through the Critical Assessment of Structure Prediction (CASP) experiments in which various prediction methods are tested over the course of a few months on sequences of proteins the 3-D structure of which is unknown at the time of the prediction ("blind" prediction). Those experiments evaluate fully automatic methods as well as human experts (see also Chapter 11 for a more detailed description of CASP and CAFASP). Due to a variety of reasons, CASP cannot be based on sufficiently large, representative data sets. Servers that automatically evaluate methods whenever new data is available address this shortcoming. Such servers base their comparisons on thousands instead of tens of test cases (as does CASP). Two such servers exist: EVA and LiveBench. EVA

[44] continuously evaluates automatic prediction methods (servers) providing results based on a large, statistically significant and, subsequently, more representative data sets. One of its principles is to facilitate comparisons on identical sets and to render comparisons on different sets very difficult. Another principle is to never distinguish in the rank between two methods if the difference in their performance is not statistically significant. Both principles are in stark contrast to what most CASP assessors did.

2.4 History

2.4.1 First Generation: Single-residue Statistics

First attempts to correlate amino acid residue frequency with secondary structure type can be traced to correlating the content of certain amino acids (e.g. proline) with the content of α-helix [176]. This was done even before the first crystallographic structures were available [81,127]. Attempts to correlate the content of all amino acids with the content of α-helix and β-strand opened the field of secondary structure prediction [19,20]. The early methods were usually based on single-residue statistics obtained from very limited data sets of known protein structures. As such they were not very accurate (Figure 2) and in addition their accuracy was overestimated at the time.

2.4.2 Second Generation: Segment Statistics

As the number of experimentally determined protein structures grew it became possible to estimate propensities for secondary structure based on consecutive segments of residues. Various numbers of adjacent residues (typically 11–21) were considered in assigning secondary structure to a central residue of a segment. Many different algorithms were applied, but they did not achieve per-residue prediction accuracies higher than slightly above 60% (Figure 2). Reports of higher accuracies were due to small data sets and did not hold for long. The main approaches used were (i) statistical information, (ii) physicochemical properties, (iii) sequence patterns, (iv) artificial neural networks, (v) graph theory, (vi) expert rules, (vii) nearest-neighbor algorithms and (viii) hybrid approaches of various algorithms.

2.4.3 Third Generation: Evolutionary Information

Proteins with similar sequences adopt similar structures [27,166]. In fact, proteins can change more than 70% of their residues without altering the basic fold [1,15,125,189]. However, the vast majority of possible sequences supposedly do not adopt globular structures at all. Rather, the exact substitution pattern of which residues can be changed and how is indicative of particular structural details. Consequently, the evolutionary information

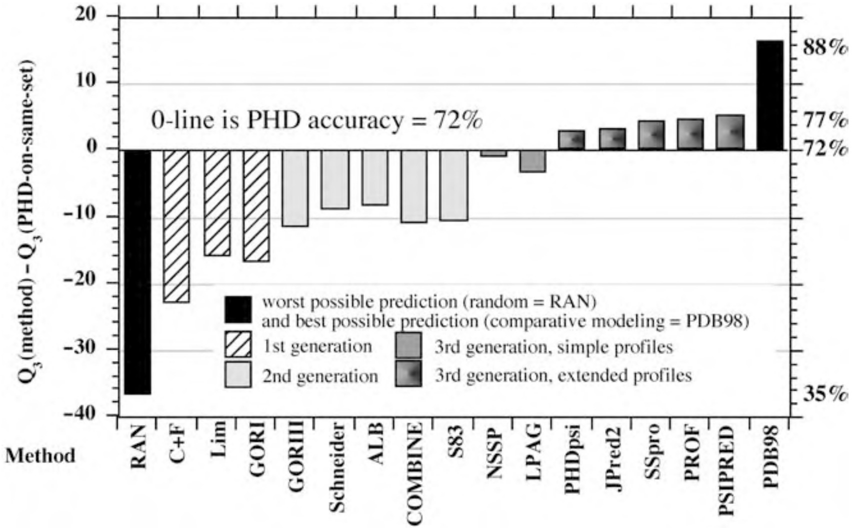


Figure 2 Three-state per-residue accuracy of various prediction methods. Included are only those methods for which we could run independent tests. Unfortunately, for most old methods this was not possible. However, for each method we had independent results from PHD (third generation, 1993) [151,154,159] available. We normalized the differences between data set by simply compiling levels of accuracy with respect to PHD. For comparison, we added the expected accuracy of a random prediction (RAN), and the best currently possible prediction accuracy achieved through comparative modeling of close homolog (PDB98). The methods were: C+F (Chou and Fasman; first generation, 1974) [28,29]; Lim (first, 1974) [93]; GORI (first, 1978) [53]; Schneider (second, 1989) [169]; ALB (second, 1983) [140]; GORIII (second, 1987) [57]; COMBINE (second, 1996) [52]; S83 (second, 1983) [77]; LPAG (third, 1993) [92]; NSSP (third, 1994) [175]; PHDpsi (third, 2001) [137]; JPred2 (third, 2000) [34]; SSpro (third, 1999) [12]; PROF (third, 2001) [149]; PSIPRED (third, 1999) [73].

contained in sequence alignments can aid structure prediction. In particular this approach improves prediction of β-strands. For the first and second generation of prediction methods β-strand prediction was particularly bad (often only slightly better than random). The pioneering method that used alignment information was proposed in the 1970s [41]. The first approaches were based on visual gathering of information from sequence alignments. In one of the first automatic algorithms making use of alignment information [107,189] the final secondary structure prediction was an average over all predictions compiled for each sequence in the alignment. The first method that succeeded in significantly improving performance by automatically using alignment information was PHD [151, 154, 157] (Figure 3). This method used a residue profile extracted from a multiple sequence alignment as an input to the artificial neural network. Many other methods used artificial neural networks [73, 123, 133], but various other algorithms were also applied successfully [38,

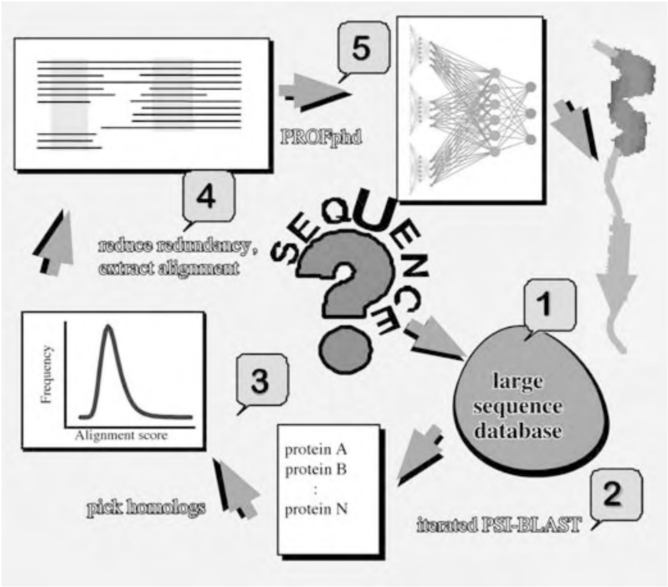


Figure 3 Using evolutionary information to predict secondary structure. Starting from a sequence of unknown structure (SEQUENCE) the following steps are required to feed evolutionary information into the PROFsec neural networks (upper right): (1 and 2) a database search for homologs through iterated PSI-BLAST [6,7] (protocol from Ref. [137]), (3) a decision for which proteins will be considered as homologs, (4) a reduction of redundancy (purge too many too similar proteins), and (5) a final refinement and extraction of the resulting multiple alignment. Numbers 1–5 illustrate where users of the PredictProtein server [151,161] can interact to improve prediction accuracy without changes made to the actual prediction method PROFsec.

39, 51, 91, 109, 146, 163] including support vector machines (SVMs) [68,181], hidden Markov models (HMMs) [79], nearest-neighbor algorithms [163].

2.4.4 **Recent Improvements of Third-generation Methods**

PHD tore down what once was a magical wall of 70% accuracy. The mark has been put much higher since. The first significant improvement was achieved by training neural networks on more diverse sequence alignments [73]. The alignments were generated by a new alignment method – PSI-BLAST [7]. It has been shown that a major improvement can be achieved by using previous types of neural networks with PSI-BLAST alignments [34]. Interestingly, it was also shown that a significant part of the improvement was simply due to the growth of sequence databases that resulted in more diverse profiles [137]. In general, the more divergent the alignment the better the prediction can be obtained. The input quality is also dependent on alignment quality. This is especially important for divergent homologous proteins where alignment methods tend to make many mistakes. Yet another simple source of improvement is related to the growth of the database of protein structures [17]. Apart from improvements in alignments, there is a lot of research pursuing development of more sophisticated and accurate algorithms. Those include new network architectures or learning techniques [3, 12, 78, 132, 133], SVMs [181] and many others.

2.4.5 **Meta-predictors Improve Somehow**

Different methods often make different mistakes. As long as those errors are not purely systematic, combining any number of methods can lead to improvements in prediction accuracy [62]. For example, the PHD method utilized this observation by combining differently trained neural networks. Various implementations of the similar concept were used in many other methods [24, 34, 128]. Alternatively, or in addition, different methods can be combined [5, 35, 36, 60, 83, 158, 170]. Overall, combinations of independent methods tend to top the single best method. However, it probably is not beneficial to use all of the available prediction methods in the meta-methods. For example, averaging over all methods evaluated by EVA evaluation server [44, 46] decreased accuracy over the best individual methods (Rost, unpublished). It is not fully straightforward how to decide whether to include a given method or not [5]. Concepts weighing the individual method based on its accuracy and "entropy" [128] appear to be successful only for large numbers of methods. More rigorous studies for the optimal combination may provide a better picture. An interesting approach resulted from attempts to improve meta-methods by developing new methods that are algorithmically different from the methods already used [85, 171]. Recently, an observation has been made indicating that optimizing meta-servers to achieve highest per-residue prediction accuracy is not always beneficial when using the final predictions in various applications [108]. Another issue that has first been introduced for secondary structure prediction is the measurement for the reliability of a prediction. To make an extreme point: a method that has 50% accuracy, but that always correctly identifies in which of the cases it is right and it which it errs (before knowing the answer), is more useful than a method with 75% accuracy and no notion about which 25% of the residues are wrong. State-of-the-art methods reliably estimate the reliability of a prediction. This is not the case for any of the existing meta-methods.

2.5 State-of-the-art Performance

2.5.1 **Average Predictions Have Good Quality**

Today's best methods reach average levels of almost 78% in Q_3 (Eq. 1) [44, 86]. They are able to accurately predict most segments (*SOV* scores around 76%).

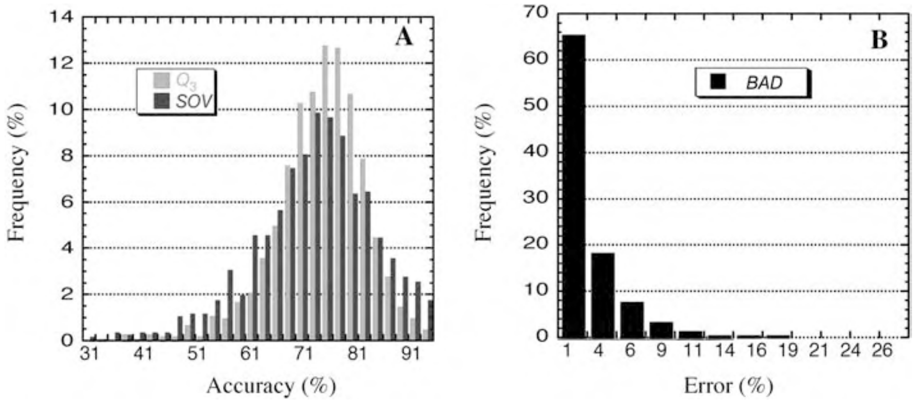


Figure 4 Expected variation of prediction accuracy for PROFsec. (A) Three-state per-residue (Q_3) and segment overlap (*SOV*) accuracies. (B) Percentage of *BAD* predictions, i.e. residues either predicted in helix and observed in strand or predicted in strand and observed in helix.

In addition the confusion between helices and strands is low (*BAD* score of less than 3%).

2.5.2 **Prediction Accuracy Varies among Proteins**

The standard deviation of three-state-per-residue accuracy computed on the per-protein basis is about 13% [44, 86] (Figure 4). Thus, some of the proteins are predicted very well (above 90%), while others are predicted very badly (even below 40% accuracy levels). The relatively large deviations are also found in prediction quality measured by other measures. The standard deviation of the *SOV* score is about 15% and that of the *BAD* score is about 5%. In particular, proteins having no sequence homologs (no alignment input) are poorly predicted. This is an important issue for the applicability of secondary structure predictions since badly predicted secondary structure is not very valuable.

2.5.3 **Reliability of Prediction Correlates with Accuracy**

For the user interested in a particular protein U, the fact that the prediction accuracy varies from protein to protein implies a rather unfortunate message: the accuracy for U could be lower than 40% or it could be higher than 90% (Figure 4). Is there any way to provide an estimate at which end of the distribution the accuracy for U is likely to be? Indeed, many methods provide numerical estimates of the expected quality of their predictions through so called reliability indices. Those indices correlate with accuracy. In other words, residues with higher reliability index are predicted with higher ac-

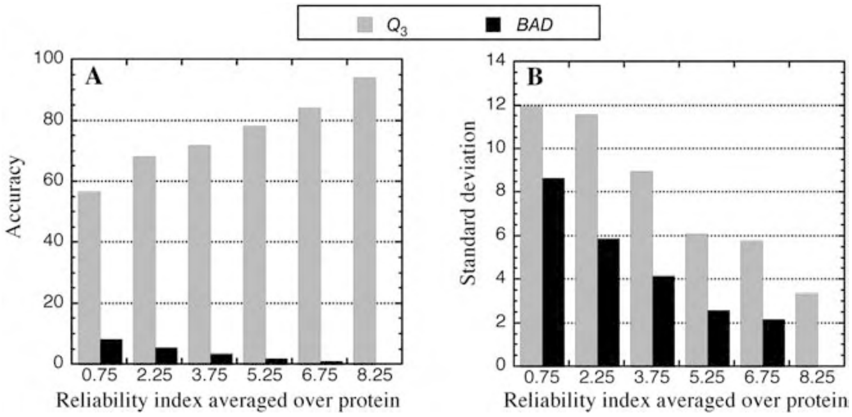


Figure 5 Prediction quality correlates with reliability indices. (A) Average three-state per-residue accuracy and *BAD* score at different reliability index thresholds (averaged over entire protein) as predicted by PROFsec [149]. (B) Corresponding values of standard deviation.

curacy [151,154,157]. Thus, the reliability index offers an excellent tool to focus on some key regions predicted at high levels of expected accuracy. Furthermore, the reliability index averaged over an entire protein correlates with the overall prediction accuracy for this protein (Figure 5).

2.5.4 **Understandable Why Certain Proteins Predicted Poorly?**

It is not easy to anticipate performance of a secondary structure prediction method based on overall structural features of proteins. However, prediction accuracy is correlated with alignment quality. Poor alignments (i.e. noninformative and/or falsely aligned residues) result in inaccurate predictions. Another interesting observation is that frequently the *BAD* predictions, i.e. the confusion between helix and strand are observed in regions that are stabilized by long-range interactions. Furthermore, helices and strands that are confused despite a high reliability index often have functional properties or are correlated to disease states (Rost, unpublished data). Regions predicted with equal propensity in two different states often correlate with “structural switches”.

2.6 Applications

2.6.1 **Better Database Searches**

Initially, three groups independently applied secondary structure predictions for fold recognition, i.e. the detection of structural similarities between proteins of unrelated sequences [50,152,162]. A few years later, almost every other fold recognition/threading method has adopted this concept [10, 37, 40, 63, 72, 74, 80, 87, 122, 124]. Two recent methods extended the concept by not only refining the database search, but by actually refining the quality of the alignment through an iterative procedure [65,71]. A related strategy has been employed to improve predictions and alignments for membrane proteins [117]. It has also been indicated that prediction mistakes tend to correlate among structurally related proteins [138], and that alignments based on purely predicted secondary structure have comparable quality with those based on matching predicted and observed states. Thus predicted secondary structure may prove useful in searching sequence databases.

2.6.2 One-dimensional Predictions Assist in the Prediction of Higher-dimensional Structure

Secondary structure predictions are now accurate enough to be used as input for methods that target the prediction of higher order aspects of protein structure automatically. A few successful applications include the following. Contact map predictions [13] have recently improved the level of accuracy significantly; an important contribution was the inclusion of secondary structure predictions [141]. They also help in the prediction of folding rates [69,142]. Secondary structure predictions have also become a popular first step toward predicting 3-D structure. Ortiz and coworkers [121] successfully use secondary structure predictions as one component of their 3-D structure prediction method. Eyrich and coworkers [47, 48] minimized the energy of arranging predicted rigid secondary structure segments. Lomize and coworkers [103] also started from secondary structure segments. Chen and coworkers [25] suggested using secondary structure predictions to reduce the complexity of molecular dynamics simulations. Levitt and coworkers [164, 165] combined secondary structure-based simplified presentations with a particular lattice simulation attempting to enumerate all possible folds.

2.6.3 Predicted Secondary Structure Helps Annotating Function

Secondary structure predictions are also useful to annotate/predict protein function. For example, secondary structure predictions have been used successfully in completely automatic predictions of subcellular localization [116]. A more typical use of secondary structure prediction is in aiding experts in finding similarities among proteins with insignificant sequence similarity. In this way functional annotation is sometimes transferred from one protein to another [184].

2.6.4 Secondary Structure-based Classifications in the Context of Genome Analysis

Proteins can be classified into families based on predicted and observed secondary structure [56, 139]. However, such procedures have been limited to a very coarse-grained grouping only sometimes useful for inferring function. Nevertheless, predictions of membrane helices and coiled-coil regions are crucial for genome analysis. More than one fifth of all eukaryotic proteins appear to have regions longer than 60 residues apparently lacking any regular secondary structure [102]. Most of these regions were not of low complexity, i.e. not composition biased. Surprisingly, these regions appeared evolutionarily as conserved as all other regions in the respective proteins. This application of secondary structure prediction may aid in classifying proteins, and in separating domains, possibly even in identifying particular functional motifs.

2.6.5 Regions Likely to Undergo Structural Change Predicted Successfully

Prions and prion-like proteins appear to aggregate through the transition of a regular secondary structure: what is "usually" a helical region switches to a strand that becomes the root of aggregation in the case of disease mutants. The reliability of the PHD secondary structure predictions combined with experimental evidence gave the first hint where this expected transition might occur [136]. Interestingly, it is still difficult to actually observe the strand in structures of even the mutant prion, while state-of-the-art prediction methods always predict the region with an observed helix to be in a strand. This example casts some light on the importance of transitions and the usefulness of predictions to capture such transitions. Young and coworkers [84] have pushed this observation further by unraveling an impressive correlation between local secondary structure predictions and global conditions. The authors monitor regions for which secondary structure prediction methods give equally strong preferences for two different states. Such regions are processed combining simple statistics and expert rules. The final method has been tested on 16 proteins known to undergo structural rearrangements and on a number of other proteins (one of those was a prion). The authors report no false positives and identify most known structural switches. Subsequently, the group applied the method to the myosin family identifying putative switching regions that were not known before, but appeared reasonable candidates [84]. This method is remarkable in two ways: (i) it is a very general method using predictions of protein structure to predict some aspects of function and (ii) it illustrates that predictions may be useful even when structures are known (as in the case of the myosin family). While the method is tailored to catch more subtle changes than occur in prions, there is some evidence that amyloid aggregation is also captured to some extent.

2.7 Things to Remember when using Predictions

2.7.1 Special Classes of Proteins

Prediction methods are usually derived from knowledge contained in proteins from subsets of current databases. Consequently, they should not be applied to classes of proteins not included in these subsets, e.g. methods for predicting helices in globular proteins are likely to fail when applied to predict transmembrane helices. In general, results should be taken with caution for proteins with unusual features, such as proline-rich regions, unusually many cysteine bonds or for domain interfaces.

2.7.2 Better Alignments Yield Better Predictions

Multiple alignment-based predictions are substantially more accurate than single sequence-based predictions [14, 39, 151]. How many sequences are needed in the alignment for an improvement; and how sensitive are prediction methods to errors in the alignment? The more sequences contained in the alignment diverge, the better (two distantly related sequences often improve secondary structure predictions by several percentage points). Regions with few aligned sequences yield less reliable predictions. The sensitivity to alignment errors depends on the methods, e.g. secondary structure prediction is less sensitive to alignment errors than solvent accessibility prediction.

2.8 Resources

2.8.1 Internet Services are Widely Available

Programs for the prediction of secondary structure available as Internet services have mushroomed since the first prediction service PredictProtein went on line in 1992 [159, 161]. The META-PredictProtein server [45] enables users to access a number of the best prediction methods through one single interface. Unfortunately, not all methods available have been sufficiently tested and some are not very accurate. This problem is addressed by the EVA server that evaluates prediction servers continuously and automatically [44, 86].

2.8.2 Interactive Services

The PHD/PROF prediction methods are automatically available via the Internet service PredictProtein [45]. Users have the choice between the fully automatic procedure taking the query sequence through the entire cycle or expert intervention into the generation of the alignment. Indeed, without spending much time users typically can improve prediction accuracy easily by choosing "good" alignments. A few of the state-of-the-art methods are also available to run locally. Note, however, that one crucial step is the generation of appropriate alignments; usually this is not "done for you" when you run the prediction method locally!

2.8.3 Servers

The following servers are publicly available (most links given by the EVA server): PROFsec [149], PHDsec [159], PHDpsi [137], PSIPRED [73], SSPRO [133], PORTER [132], SABLE [3], SAM-T02 [79], Jpred [34], APSSP, JUFO [110], PROF [123], YASPIN [94].

3 Transmembrane Regions

3.1 Transmembrane Proteins are an Extremely Important Class of Proteins

Approximately 15–30% of all proteins are estimated to contain transmembrane regions [97, 111]. Those proteins are responsible for the communication between the cell and its surroundings, and are of great importance to biomedicine. The cell membrane environment, composed of a lipid bilayer, is very different from one found in most cellular compartments. The transmembrane segments of proteins tend to be hydrophobic which enables them to remain within a membrane by avoiding the solvent present at both boundaries. The special features of transmembrane protein sequences serve as the basis for identifying them by computational methods. As in case of globular proteins, the transmembrane segments form regular secondary structures and can be assigned to two broad classes: those composed entirely of helices and those composed of strands (despite ardent searches and putative evidence, we still do not have any proof for the existence of a mixture of the two). By far the majority of all membrane proteins appear to be of the helical type [18]. An important characteristic of transmembrane proteins is the orientation of membrane segments with respect to the N-terminus of a protein, often referred to as the topology. Usually, the successful prediction of transmembrane segments requires proper identification of transmembrane regions in sequence, actual prediction of the secondary structure and deciphering the topology. It is very difficult to experimentally determine 3-D structures for transmembrane proteins. Despite considerable advances over the last decade, we still have experimental structures or theoretical models for supposedly less than 10% of, for example, all human membrane proteins (Punta, Liu and Rost, unpublished). Useful predictions of structural and functional aspects are therefore highly needed.

3.2 Prediction Methods

Although all known transmembrane regions constitute of regular secondary structures, most secondary structure prediction methods developed for nonmembrane proteins mostly fail to correctly predict membrane regions. Furthermore, very few methods have been developed for proteins with β-strands in the membrane. The first and most basic methods for helical membrane regions focused on identification of transmembrane segments based simply on residue hydrophobicity [90]. It was observed that positively charged residues are more abundant on the inside of the membrane (the "positive-in" rule). A simple Kyte–Doolittle hydrophobicity plot [90] can thus provide much information on the presence of such transmembrane segments. This led to the development of the method that predicted positions of helices and the topology of helical membrane proteins [179]. Next, neural networks were applied to better identify transmembrane helices, and differentiate between membrane and nonmembrane proteins [153]. Among other approaches were HMM methods attempting to match the sequence to the predefined "grammar" of transmembrane proteins [88] (see Chapter 3 for basics on HMMs) and many others [33,66]. Recently, groups have begun to venture into the development of methods that predict membrane regions with β-strands [18, 42, 59, 70].

3.3 Performance

Estimates for the accuracy in predicting membrane regions are extremely problematic because there are so few high-resolution structures available. Consequently, all methods in the past were evaluated by also using low-resolution information from biochemical experiments that provide some evidence for the location of transmembrane regions. Unfortunately, such experiments can be more inaccurate than prediction methods [26]. This was one of the reasons why the performance of prediction methods had been significantly overestimated by the end of the last millennium [26, 113]. It now appears that the best prediction methods correctly predict all membrane helices for about 50–70% of all proteins, very few methods avoid the confusion between very hydrophobic signal peptides and membrane proteins, and the best methods falsely identify membrane helices in about 10% of all nonmembrane proteins [26, 113]. However, results can be far worse, e.g. most hydrophobicity-based methods misclassify over 50% (!) of all globular proteins as "containing membrane helices" [26]. Overestimates in publications are also a very serious problem – even over the last few years, methods have been published in prominent journals with estimated levels of above 95% accuracy that failed to reach significantly above 50% and misclassified over 30% of the globular proteins [26]. Note also that there are a few top methods available at the moment; all of these have their own strengths and weaknesses, i.e. there is no single one “best method”. Predictions of β-barrel membrane regions currently appear to be more accurate than those for helical membrane regions; however, this may likely turn out to be an overestimate caused by the fact that we have too limited experimental information.

3.4 Servers

There are many more methods than the following available; however, the methods listed here have sustained many evaluations. Helical membrane proteins: PHDhtm [153], SOSUI [66], TopPred [179], TMHMM [88], DAS [33]; β-barrel membrane regions: ProfTMB [18].

4 Solvent Accessibility

4.1 Solvent Accessibility Somehow Distinguishes Structurally Important from Functionally Important

In 3-D structures of globular proteins some of residues are buried deep inside, whereas others are located on the surface and thus are more exposed to the surrounding solvent. Residues that are more exposed to solvent are also more accessible to other biological agents and, consequently, are much more likely to be involved in functional interactions which require spatial accessibility such as enzymatic activity, DNA binding, signal transduction, etc. However, buried residues are much more likely to play important roles in stabilizing structures of proteins. Thus, a good distinction between exposed and buried residues can be very useful to distinguish residues that are important for function (conserved and exposed) from those that are important for structure (conserved and buried).

4.2 Measuring Solvent Accessibility

Solvent accessibility is usually measured in terms of the surface area accessible to water molecules. The values can range from 0 Å for entirely buried residues to around 300 Å for the largest residues on the surfaces of proteins. A measure that is not dependent on the size of the amino acid residue is the relative solvent accessibility expressed as a percentage of the residue surface that is exposed to solvent. It appears that among homologous proteins the relative solvent accessibility is less conserved than secondary structures [155]. In addition, the solvent accessibility of protein residues is strongly influenced by nonlocal interactions, where residues located far away along a protein sequence can be in spatial proximity resulting in mutual screening from solvent. Thus, predicting solvent accessibility appears to be more difficult than prediction of secondary structure. It addition, it was shown that among the evolutionarily related proteins of similar structure buried residues (less than 10% accessible surface area) tend to be much more highly conserved than highly exposed residues (more than 60%) [155]. Thus, for methods that use evolutionary information derived from alignments of related proteins it should be easier to closely predict accessibility for buried residues than for the exposed ones. A simplified approach is to try to distinguish between residues below a certain solvent accessibility threshold ("buried") and those above it ("exposed"). There is no biophysical reason to choose one threshold over another, and different researchers often choose different thresholds (7, 9, 16 and 25% are used). On average, about half of all protein residues have more than 25% of their surfaces exposed.

4.3 Best Methods Combine Evolutionary Information with Machine Learning

Some of the methods that predict secondary structure also have the capability of predicting solvent accessibility, since essentially the same basic concepts apply to building a solvent accessibility predictor. For example, PHDacc [155] and PROFacc [149] methods, which are part of the PredictProtein [159, 161] server, use the same sequence profile input as do their respective secondary structure prediction counterparts (PHDsec and PROFsec). They use a neural network that assigns relative solvent accessibility into one of the 10 states corresponding to squares of relative solvent accessibility (state 10 corresponds to a range 81–100% of solvent accessibility). This 10-state scheme can be converted to a two-state scheme or to a prediction in terms of actual value of the exposed surface. Another well known method is Jpred [36]. It is also a server that predicts both secondary structure and solvent accessibility. The method uses alignments generated by HMMs and PSI-BLAST as input to a neural network. The output of predictions from two different networks is combined to give a final relative solvent accessibility. Many other variations and similar approaches have been attempted which include various types of neural networks [2, 4, 34, 131], SVMs [82], Bayesian networks [177], information-theoretic approaches [115] and simple baseline approaches [144]. Most recently the relation between secondary structure and accessibility was explored to develop methods that combine both predictions explicitly to improve each one [2, 149].

4.4 Performance

Unlike the prediction of secondary structure that is continuously assessed and monitored on identical data sets, methods for the prediction of solvent accessibility are not. Given that different groups use widely different data sets and different conventions to convert actual values of solvent accessibility into prediction states, it is impossible to compare and reasonably summarize levels of performance. However, two-state predictions (either buried or exposed) are predicted at levels above 75% accuracy. Whatever values you read, note that advanced methods are significantly more accurate than simple methods based on simple features such as hydrophobicity, polarity or simple statistics.

4.5 Servers

PROFacc [149], PHDacc [155], SABLE [2], Jpred [34], ACCpro [131].

5 Inter-residue Contacts

5.1 Two-dimensional Predictions may be a Step Toward 3-D Structures

Directly predicting 3-D structure still fails. Predictions of 1-D aspects of protein structure, such as secondary structure and solvent accessibility, provide very valuable information. However, 1-D predictions are far too simplified. There is a path seemingly in between these two extremes (1-D/3-D), i.e. the prediction of inter-residue distances. In fact, 3-D structures can be reconstructed more or less completely from 2-D distance maps. The catch is that distance maps are as hard to guess as 3-D coordinates. As a consequence, existing methods try to solve the simplified problem of predicting contact maps, where two residues are considered to be in contact if they are located within a certain spatial cut-off distance (this results in a binary classification of residue pairs, i.e. contact/noncontact pairs).

5.2 Measuring Performance

There is no widely accepted threshold for the maximal distance between two residues that are considered as "in contact". While the smallest physically possible distance could be agreed upon, the limit beyond which the interaction between two residues can be considered negligible is more difficult to define. However, the distance of 8 Å between C_β atoms is the most widely used threshold for the evaluation of the performance of these prediction methods. The output of contact prediction programs is generally a list of residue pairs, ranked according to some internal confidence score. Usually, only contacts between pairs that exceed a minimal sequence separation are evaluated. Although many different thresholds have been used, minimal separations of six and 24 sequence positions are most common for prediction of medium- and long-range contacts, respectively. These parameters are important as the task becomes more difficult with increasing separation (this tendency levels off for separations over 20).

5.3 Prediction Methods

One line of methods was based upon the observation that evolutionary pressure on maintaining protein structure would sometimes require correlation in the mutations of amino acid residues that are in spatial proximity to each other. In principle, such patterns of correlation could be discerned in the multiple alignments of protein sequences. Some of the early contact prediction methods have indeed used only correlated mutations computed from multiple sequence alignments [58, 119]. The currently best methods make also use of other protein features, such as evolutionary profiles of the nearest neighbors of the residue pair being predicted, sequence separation, secondary structure and solvent accessibility predictions. Further improvement of predictions was achieved through machine learning techniques such as: neural networks that use [49, 61, 120] or do not use [130, 141] correlated mutations, HMMs [22, 129, 172], SVMs [188] and genetic programming [104].

5.4 Performance and Applications

As the prediction of nonlocal contacts is difficult, progress in the field had been slow until recently when two promising new methods entered the CASP6 competition in 2004. When $L/2$ predictions are considered, the accuracy of state-of-the-art methods is around 30% for sequence separation of at least six and around 20% for sequence separation of at least 24. Although predicted contact maps are not very accurate, they are nevertheless better on average than the contact maps obtained from the best *de novo* predictions of 3-D structures [46]. As a result, the automatically predicted contact maps were successfully used in prediction of 3-D protein structures [119, 121, 174].

5.5 Servers

PROFcon [141], CORNET [120], CMAPpro [129], GPCPRED [104], Hamilton's server [61].

6 Flexible and Intrinsically Disordered Regions

6.1 Local Mobility, Rigidity and Disorder all are Features that Relate to Function

In crystal structures of proteins, the uncertainty of atomic positions can be represented by B-factors (Debye–Waller factors) [32]. B-factors represent the combined effects of thermal variation and static disorder. In general, the higher the B-factor of a residue, the higher is its flexibility. Further, it has been demonstrated that many proteins and protein regions lack a unique 3-D structure [180]. Those regions are often characterized as an ensemble of rapidly changing alternative structures with differing backbone torsion angles. Estimates indicate that a substantial fraction of all proteins (as much as 25%) may contain disordered regions or be entirely disordered [43, 102, 148, 182]. Many important functional interactions, such as cell-cycle regulation, signal transduction, gene expression and chaperon action, are associated with proteins containing very flexible and disordered regions. Determination of these regions also plays an important role in structural genomics, since such regions can be a source of problems in protein expression, purification and crystallization.

6.2 Measuring Flexibility and Disorder

Protein flexibility can be derived from normalized B-factors [23]. Characterization of disordered regions can be provided by many experimental techniques, but in particular by NMR spectroscopy. Regions of protein X-ray structures without atomic coordinates are often considered as intrinsically disordered regions. Successful predictions should be able to simply indicate intrinsically disordered regions, or in case of protein flexibility to assign accurate normalized B-factors to protein residues.

6.3 Prediction Methods

Methods predicting regions of low compositional complexity in protein sequences (SEG [185] and CAST [135]) can be considered as the first methods predicting disordered regions in proteins. However, the correlation between low-complexity regions and disorder is far from perfect. The low-complexity regions are highly repetitive in their amino acid composition but many of them have well defined 3-D structures [167]. There are methods that attempt to predict if entire proteins are in "natively unfolded" configurations based on hydrophobicity and charge information derived from sequences [178]. The disordered regions can be predicted based on disorder propensity assigned to each amino acid [95]. Other methods use machine learning algorithms such as neural networks [75, 95, 147] or SVMs [182]. The NORSp method [99] predicts extended nonregular secondary structure segments that often correlate with disorder. Predictions of B-factors were also carried out by methods using artificial neural networks [168] or support vector regression [186]. The prediction accuracy of those methods was not experimentally verified on the large scale yet.

6.4 Servers

PROFbval [168], PONDR [147], DISOPRED [75], DISOPRED2 [182], GlobPlot [95], NORSp [99], FoldIndex [134], DisEMBL [95].

7 Protein Domains

7.1 Independent Folding Units

The visual inspection of 3-D structures of large proteins often reveals compact structural subunits referred to as protein domains. Such domains are assumed to often constitute units that fold independently. Studies indicate that some of those proteins can be viewed as combinatorial arrangements of protein domains that are genetically mobile. Often, the structural domains are associated with particular biological functions. It is postulated that domains are independent folding units of large proteins. Knowledge of the domain organization of proteins of unknown 3-D structures can help experimental and computational attempts to elucidate their structure and function. Recent analyses of sequence-structure families suggest that over two-thirds of all proteins have more than one domain and that most domains span over about 100 residues [96].

7.2 Prediction Methods

The prediction of the domain organization is a challenging problem if we do not know the 3-D structure (and automatic assignment methods disagree much more than secondary structure assignment methods even if we know the structure). Many sequence-based methods predict domains that are significantly shorter than actual structural domains [98]. The first automatic prediction methods, such as ProDom [31], attempted to determine domains based on “boundaries” in multiple alignments of protein sequences. This approach often results in fragmentation of actual structural domains since sequence similarity conservation often does not extend over entire domains.

In a similar approach, domain constraints can also be obtained from sequence alignment databases such as BLOCKS [64]. Attempts to explicitly elongate sequence alignments were also made [54]. Other automatic prediction methods apply concepts from protein structure prediction [55] or try do derive domains from predicted contact maps [145]. There are methods that use statistics of domain size distributions [183] or a statistical approach toward combining various sources of information [89]. Some of the methods use artificial neural networks [112,114]. Others explore alternative ways of using sequence alignment information [105] or alignments of predicted secondary structure elements [106]. The most accurate methods (e.g. CHOP [96]) simply use sequence homology to proteins with known domain assignments. The downside of such methods is the low coverage, i.e. that they often do not find domains. None of these more recent methods has yet been experimentally verified on large scale.

7.3 Servers

CHOP (homology based) [96], CHOPnet [100], ProDom (homology based) [31], DOMAINATION [54], SnapDRAGON [55], DomSSEA [106].

Acknowledgments

We are grateful to Kaz Wrzeszczynski, Marco Punta and Avner Schlessinger (all from Columbia University) for valuable input. Thanks to Volker Eyrich (Schroedinger Inc.) and Ingrid Koh (Columbia) for their help in setting up the EVA server. The work of D. P. and B. R. was supported by the grant RO1-LM07329-01 from the National Library of Medicine. Last, not least, thanks to all those who deposit their experimental data in public databases and to those who maintain these databases.

References

1 Abagyan, R. A. and S. Batalov. 1997. Do aligned sequences share the same fold? J. Mol. Biol. **273**: 355–68.

2 Adamczak, R., A. Porollo and J. Meller. 2004. Accurate prediction of solvent accessibility using neural networks-based regression. Proteins **56**: 753–67.

3 Adamczak, R., A. Porollo and J. Meller. 2005. Combining prediction of secondary structure and solvent accessibility in proteins. Proteins **59**: 467–75.

4 Ahmad, S. and M. M. Gromiha. 2002. NETASA: neural network based prediction of solvent accessibility. Bioinformatics **18**: 819–24.

5 Albrecht, M., S. C. Tosatto, T. Lengauer and G. Valle. 2003. Simple consensus procedures are effective and sufficient in secondary structure prediction. Protein Eng. **16**: 459–62.

6 ALTSCHUL, S. F. AND W. GISH. 1996. Local alignment statistics. Methods Enzymol. **266**: 460–80.

7 ALTSCHUL, S. F., T. L. MADDEN, A. A. SCHAEFFER, J. ZHANG, Z. ZHANG, W. MILLER AND D. J. LIPMAN. 1997. Gapped Blast and PSI-Blast: a new generation of protein database search programs. Nucleic Acids Res. **25**: 3389–402.

8 ANDERSEN, C. A. F., A. G. PALMER, S. BRUNAK AND B. ROST. 2002. Continuum secondary structure captures protein flexibility. Structure **10**: 175–84.

9 ANFINSEN, C. B. 1973. Principles that govern the folding of protein chains. Science **181**: 223–30.

10 AYERS, D. J., P. R. GOOLEY, A. WIDMER-COOPER AND A. E. TORDA. 1999. Enhanced protein fold recognition using secondary structure information from NMR. Protein Sci. **8**: 1127–33.

11 BAIROCH, A. AND R. APWEILER. 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. Nucleic Acids Res. **28**: 45–8.

12 BALDI, P., S. BRUNAK, P. FRASCONI, G. SODA AND G. POLLASTRI. 1999. Exploiting the past and the future in protein secondary structure prediction. Bioinformatics **15**: 937–46.

13 BALDI, P., G. POLLASTRI, C. A. ANDERSEN AND S. BRUNAK. 2000. Matching protein beta-sheet partners by feedforward and recurrent neural networks. Proc. ISMB **8**: 25–36.

14 BARTON, G. J. 1995. Protein secondary structure prediction. Curr. Opin. Struct. Biol. **5**: 372–76.

15 BENNER, S. A. AND D. GERLOFF. 1991. Patterns of divergence in homologous proteins as indicators of secondary and tertiary structure: a prediction of the structure of the catalytic domain of protein kinases. Adv. Enzyme Regul. **31**: 121–81.

16 BENSON, D. A., I. KARSCH-MIZRACHI, D. J. LIPMAN, J. OSTELL AND D. L. WHEELER. 2003. GenBank. Nucleic Acids Res. **31**: 23–27.

17 BERMAN, H. M., J. WESTBROOK, Z. FENG, G. GILLLILAND, T. N. BHAT, H. WEISSIG, I. N. SHINDYALOV AND P. E. BOURNE. 2000. The Protein Data Bank. Nucleic Acids Res. **28**: 235–42.

18 BIGELOW, H. R., D. S. PETREY, J. LIU, D. PRZYBYLSKI AND B. ROST. 2004. Predicting transmembrane beta-barrels in proteomes. Nucleic Acids Res. **32**: 2566–77.

19 BLOUT, E. R. 1962. The dependence of the conformation of polypetides and proteins upon amino acid composition. In STAHMAN, M. (ed.), *Polyamino Acids, Polypeptides, and Proteins*. University of Wisconsin Press, Madison, WI: 275–79.

20 BLOUT, E. R., C. DE LOZÉ, S. M. BLOOM AND G. D. FASMAN. 1960. Dependence of the conformation of synthetic polypeptides on amino acid composition. J. Am. Chem. Soc. **82**: 3787–9.

21 BUSSIERE, D. E., X. KONG, D. A. EGAN, K. WALTER, T. F. HOLZMAN, F. LINDH, T. ROBINS AND V. L. GIRANDA. 1998. Structure of the E2 DNA-binding domain from human papillomavirus serotype 31 at 2.4 Å. Acta Crystallogr. D **54**: 1367–76.

22 BYSTROFF, C. AND Y. SHAO. 2002. Fully automated ab initio protein structure prediction using I-SITES, HMMSTR and ROSETTA. Bioinformatics **18**: S54–61.

23 CARUGO, O. AND P. ARGOS. 1997. Correlation between side chain mobility and conformation in protein structures. Protein Eng. **10**: 777–87.

24 CHANDONIA, J. M. AND M. KARPLUS. 1999. New methods for accurate prediction of protein secondary structure. Proteins **35**: 293–306.

25 CHEN, C. C., J. P. SINGH AND R. B. ALTMAN. 1999. Using imperfect secondary structure predictions to improve molecular structure computations. Bioinformatics **15**: 53–65.

26 CHEN, C. P., A. KERNYTSKY AND B. ROST. 2002. Transmembrane helix predictions revisited. Protein Sci. **11**: 2774–91.

27 CHOTHIA, C. AND A. M. LESK. 1986. The relation between the divergence of sequence and structure in proteins. EMBO J. **5**: 823–26.

28 Chou, P. Y. and G. D. Fasman. 1974. Prediction of protein conformation. Biochemistry **13**: 211–5.

29 Chou, P. Y. and G. D. Fasman. 1978. Prediction of the secondary structure of proteins from their amino acid sequence. Adv. Enzymol. **47**: 45–148.

30 Colloc'h, N., C. Etchebest, E. Thoreau, B. Henrissat and J. P. Mornon. 1993. Comparison of three algorithms for the assignment of secondary structure in proteins: the advantages of a consensus assignment. Protein Eng. **6**: 377–82.

31 Corpet, F., F. Servant, J. Gouzy and D. Kahn. 2000. ProDom and ProDom-CG: tools for protein domain analysis and whole genome comparisons. Nucleic Acids Res. **28**: 267–9.

32 Creighton, T. 1992. *Proteins: Structures and Molecular Properties*. Freeman, San Francisco, CA.

33 Cserzo, M., F. Eisenhaber, B. Eisenhaber and I. Simon. 2004. TM or not TM: transmembrane protein prediction with low false positive rate using DAS-TMfilter. Bioinformatics **20**: 136–7.

34 Cuff, J. A. and G. J. Barton. 2000. Application of multiple sequence alignment profiles to improve protein secondary structure prediction. Proteins **40**: 502–11.

35 Cuff, J. A. and G. J. Barton. 1999. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins **34**: 508–19.

36 Cuff, J. A., M. E. Clamp, A. S. Siddiqui, M. Finlay and G. J. Barton. 1998. JPred: a consensus secondary structure prediction server. Bioinformatics **14**: 892–3.

37 de la Cruz, X. and J. M. Thornton. 1999. Factors limiting the performance of prediction-based fold recognition methods. Protein Sci. **8**: 750–9.

38 Defay, T. and F. E. Cohen. 1995. Evaluation of current techniques for *ab initio* protein structure prediction. Proteins **23**: 431–45.

39 Di Francesco, V., J. Garnier and P. J. Munson. 1996. Improving protein secondary structure prediction with aligned homologous sequences. Protein Sci. **5**: 106–13.

40 Di Francesco, V., P. J. Munson and J. Garnier. 1999. FORESST: fold recognition from secondary structure predictions of proteins. Bioinformatics **15**: 131–40.

41 Dickerson, R. E., R. Timkovich and R. J. Almassy. 1976. The cytochrome fold and the evolution of bacterial energy metabolism. J. Mol. Biol. **100**: 473–91.

42 Diederichs, K., J. Freigang, S. Umhau, K. Zeth and J. Breed. 1998. Prediction by a neural network of outer membrane beta-strand protein topology. Protein Sci. **7**: 2413–20.

43 Dunker, A. K., C. J. Brown, J. D. Lawson, L. M. Iakoucheva and Z. Obradovic. 2002. Intrinsic disorder and protein function. Biochemistry **41**: 6573–82.

44 Eyrich, V. A., M. A. Martí-Renom, D. Przybylski, A. Fiser, F. Pazos, A. Valencia, A. Sali and B. Rost. 2001. EVA: continuous automatic evaluation of protein structure prediction servers. Bioinformatics **17**: 1242–3.

45 Eyrich, V. A. and B. Rost. 2003. META-PP: single interface to crucial prediction servers. Nucleic Acids Res. **31**: 3308–10.

46 Eyrich, V. A., D. Przybylski, I. Y. Koh, O. Grana, F. Pazos, A. Valencia and B. Rost. 2003. CAFASP3 in the spotlight of EVA. Proteins **53 (Suppl. 6)**: 548–60.

47 Eyrich, V. A., D. M. Standley, A. K. Felts and R. A. Friesner. 1999. Protein tertiary structure prediction using a branch and bound algorithm. Proteins **35**: 41–57.

48 Eyrich, V. A., D. M. Standley and R. A. Friesner. 1999. Prediction of protein tertiary structure to low resolution: performance for a large and structurally diverse test set. J. Mol. Biol. **288**: 725–42.

49 Fariselli, P., O. Olmea, A. Valencia and R. Casadio. 2001. Prediction of contact maps with neural networks and correlated mutations. Protein Eng. **14**: 835–43.

50 Fischer, D. and D. Eisenberg. 1996. Fold recognition using sequence-derived properties. Protein Sci. **5**: 947–55.

51 Frishman, D. and P. Argos. 1995. Knowledge-based protein secondary structure assignment. Proteins **23**: 566–79.

52 Garnier, J., J.-F. Gibrat and B. Robson. 1996. GOR method for predicting protein secondary structure from amino acid sequence. Methods Enzymol. **266**: 540–53.

53 Garnier, J., D. J. Osguthorpe and B. Robson. 1978. Analysis of the accuracy and Implications of simple methods for predicting the secondary structure of globular proteins. J. Mol. Biol. **120**: 97–120.

54 George, R. A. and J. Heringa. 2002. Protein domain identification and improved sequence similarity searching using PSI-BLAST. Proteins **48**: 672–81.

55 George, R. A. and J. Heringa. 2002. SnapDRAGON: a method to delineate protein structural domains from sequence data. J. Mol. Biol. **316**: 839–51.

56 Gerstein, M. and M. Levitt. 1997. A structural census of the current population of protein sequences. Proc. Natl Acad. Sci. USA **94**: 11911–6.

57 Gibrat, J.-F., J. Garnier and B. Robson. 1987. Further developments of protein secondary structure prediction using information theory. New parameters and consideration of residue pairs. J. Mol. Biol. **198**: 425–43.

58 Gobel, U., C. Sander, R. Schneider and A. Valencia. 1994. Correlated mutations and residue contacts in proteins. Proteins **18**: 309–17.

59 Gromiha, M. M., R. Majumdar and P. K. Ponnuswamy. 1997. Identification of membrane spanning beta strands in bacterial porins. Protein Eng. **10**: 497–500.

60 Guermeur, Y., C. Geourjon, P. Gallinari and G. Deleage. 1999. Improved performance in protein secondary structure prediction by inhomogeneous score combination. Bioinformatics **15**: 413–21.

61 Hamilton, N., K. Burrage, M. A. Ragan and T. Huber. 2004. Protein contact prediction using patterns of correlation. Proteins **56**: 679–84.

62 Hansen, L. K. and P. Salamon. 1990. Neural network ensembles. IEEE Trans. Pattern Anal. Machine Intell. **12**: 993–1001.

63 Hargbo, J. and A. Elofsson. 1999. Hidden Markov models that use predicted secondary structures for fold recognition. Proteins **36**: 68–76.

64 Henikoff, J. G. and S. Henikoff. 1996. Blocks database and its applications. Methods Enzymol. **266**: 88–105.

65 Heringa, J. 1999. Two strategies for sequence comparison: profile-preprocessed and secondary structure-induced multiple alignment. Comput. Chem. **23**: 341–64.

66 Hirokawa, T., S. Boon-Chieng and S. Mitaku. 1998. SOSUI: classification and secondary structure prediction system for membrane proteins. Bioinformatics **14**: 378–79.

67 Hobohm, U., M. Scharf, R. Schneider and C. Sander. 1992. Selection of representative protein data sets. Protein Sci. **1**: 409–17.

68 Hua, S. and Z. Sun. 2001. A novel method of protein secondary structure prediction with high segment overlap measure support vector machine approach. J. Mol. Biol. **308**: 397–407.

69 Ivankov, D. N. and A. V. Finkelstein. 2004. Prediction of protein folding rates from the amino acid sequence-predicted secondary structure. Proc. Natl Acad. Sci. USA **101**: 8942–4.

70 Jacoboni, I., P. L. Martelli, P. Fariselli, V. De Pinto and R. Casadio. 2001. Prediction of the transmembrane regions of beta-barrel membrane proteins with a neural network-based predictor. Protein Sci. **10**: 779–87.

71 Jennings, A. J., C. M. Edge and M. J. Sternberg. 2001. An approach to improving multiple alignments of protein sequences using predicted secondary structure. Protein Eng. **14**: 227–31.

72 Jones, D. T. 1999. GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. J. Mol. Biol. **287**: 797–815.

73 Jones, D. T. 1999. Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. **292**: 195–202.

74 Jones, D. T., M. Tress, K. Bryson and C. Hadley. 1999. Successful recognition of protein folds using threading methods biased by sequence similarity and predicted secondary structure. Proteins **37**: 104–11.

75 Jones, D. T. and J. J. Ward. 2003. Prediction of disordered regions in proteins from position specific score matrices. Proteins **53 (Suppl. 6)**: 573–8.

76 Kabsch, W. and C. Sander. 1983. Dictionary of protein secondary structure: pattern recognition of hydrogen bonded and geometrical features. Biopolymers **22**: 2577–637.

77 Kabsch, W. and C. Sander. 1983. How good are predictions of protein secondary structure? FEBS Lett. **155**: 179–82.

78 Karchin, R., M. Cline, Y. Mandel-Gutfreund and K. Karplus. 2003. Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. Proteins **51**: 504–14.

79 Karplus, K., R. Karchin, J. Draper, J. Casper, Y. Mandel-Gutfreund, M. Diekhans and R. Hughey. 2003. Combining local-structure, fold-recognition, and new fold methods for protein structure prediction. Proteins **53**: 491–6.

80 Kelley, L. A., R. M. MacCallum and M. J. Sternberg. 2000. Enhanced genome annotation using structural profiles in the program 3D-PSSM. J. Mol. Biol. **299**: 499–520.

81 Kendrew, J. C., R. E. Dickerson, B. E. Strandberg, R. J. Hart, D. R. Davies and D. C. Phillips. 1960. Structure of myoglobin: a three-dimensional Fourier synthesis at 2 Å resolution. Nature **185**: 422–7.

82 Kim, H. and H. Park. 2004. Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3D local descriptor. Proteins **54**: 557–62.

83 King, R. D., M. Ouali, A. T. Strong, A. Aly, A. Elmaghraby, M. Kantardzic and D. Page. 2000. Is it better to combine predictions? Protein Eng. **13**: 15–9.

84 Kirshenbaum, K., M. Young and S. Highsmith. 1999. Predicting allosteric switches in myosins. Protein Sci. **8**: 1806–15.

85 Kloczkowski, A., K. L. Ting, R. L. Jernigan and J. Garnier. 2002. Combining the GOR V algorithm with evolutionary information for protein secondary structure prediction from amino acid sequence. Proteins **49**: 154–66.

86 Koh, I. Y., V. A. Eyrich, M. A. Marti-Renom, et al. 2003. EVA: evaluation of protein structure prediction servers. Nucleic Acids Res. **31**: 3311–5.

87 Koretke, K. K., R. B. Russell, R. R. Copley and A. N. Lupas. 1999. Fold recognition using sequence and secondary structure information. Proteins **37**: 141–8.

88 Krogh, A., B. Larsson, G. von Heijne and E. L. Sonnhammer. 2001. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. J. Mol. Biol. **305**: 567–80.

89 Kulikowski, C. A., I. Muchnik, H. J. Yun, A. A. Dayanik, D. Zhang, Y. Song and G. T. Montelione. 2001. Protein structural domain parsing by consensus reasoning over multiple knowledge sources and methods. Medinfo **10**: 965–9.

90 Kyte, J. and R. F. Doolittle. 1982. A simple method for displaying the hydrophathic character of a protein. J. Mol. Biol. **157**: 105–32.

91 Levin, J. M. 1997. Exploring the limits of nearest neighbour secondary structure prediction. Protein Eng. **10**: 771–6.

92 Levin, J. M., S. Pascarella, P. Argos and J. Garnier. 1993. Quantification of secondary structure prediction improvement using multiple alignment. Protein Eng. **6**: 849–54.

93 Lim, V. I. 1974. Structural principles of the globular organization of protein chains. a stereochemical theory of globular protein secondary structure. J. Mol. Biol. **88**: 857–72.

94 LIN, K., V. A. SIMOSSIS, W. R. TAYLOR AND J. HERINGA. 2005. A simple and fast secondary structure prediction method using hidden neural networks. Bioinformatics **21**: 152–9.

95 LINDING, R., R. B. RUSSELL, V. NEDUVA AND T. J. GIBSON. 2003. GlobPlot: Exploring protein sequences for globularity and disorder. Nucleic Acids Res. **31**: 3701–8.

96 LIU, J. AND B. ROST. 2004. CHOP: parsing proteins into structural domains. Nucleic Acids Res. **32**: W569–71.

97 LIU, J. AND B. ROST. 2001. Comparing function and structure between entire proteomes. Protein Sci. **10**: 1970–9.

98 LIU, J. AND B. ROST. 2003. Domains, motifs and clusters in the protein universe. Curr. Opin. Chem. Biol. **7**: 5–11.

99 LIU, J. AND B. ROST. 2003. NORSp: predictions of long regions without regular secondary structure. Nucleic Acids Res. **31**: 3833–5.

100 LIU, J. AND B. ROST. 2004. Sequence-based prediction of protein domains. Nucleic Acids Res. **32**: 3522–30.

101 LIU, J. AND B. ROST. 2002. Target space for structural genomics revisited. Bioinformatics **18**: 922–33.

102 LIU, J., H. TAN AND B. ROST. 2002. Loopy proteins appear conserved in evolution. J. Mol. Biol. **322**: 53–64.

103 LOMIZE, A. L., I. D. POGOZHEVA AND H. I. MOSBERG. 1999. Prediction of protein structure: the problem of fold multiplicity. Proteins **Suppl. 3**: 199–203.

104 MACCALLUM, R. M. 2004. Striped sheets and protein contact prediction. Bioinformatics **20 (Suppl. 1)**: i224–31.

105 MARCOTTE, E. M., M. PELLEGRINI, M. J. THOMPSON, T. O. YEATES AND D. EISENBERG. 1999. A combined algorithm for genome-wide prediction of protein function. Nature **402**: 83–6.

106 MARSDEN, R. L., L. J. MCGUFFIN AND D. T. JONES. 2002. Rapid protein domain assignment from amino acid sequence using predicted secondary structure. Protein Sci. **11**: 2814–24.

107 MAXFIELD, F. R. AND H. A. SCHERAGA. 1979. Improvements in the prediction of protein topography by reduction of statistical errors. Biochemistry **18**: 697–704.

108 MCGUFFIN, L. J. AND D. T. JONES. 2003. Benchmarking secondary structure prediction for fold recognition. Proteins **52**: 166–75.

109 MEHTA, P. K., J. HERINGA AND P. ARGOS. 1995. A simple and fast approach to prediction of protein secondary structure from multiply aligned sequences with accuracy above 70%. Protein Sci. **4**: 2517–25.

110 MEILER, J., M. MUELLER, A. ZEIDLER AND F. SCHMAESCHKE. 2001. Generation and evaluation of dimension-reduced amino acid parameter representation by artificial neural networks. J. Mol. Model. **7**: 360–9.

111 MELEN, K., A. KROGH AND G. VON HEIJNE. 2003. Reliability measures for membrane protein topology prediction algorithms. J. Mol. Biol. **327**: 735–44.

112 MIYAZAKI, S., Y. KURODA AND S. YOKOYAMA. 2002. Characterization and prediction of linker sequences of multi-domain proteins by a neural network. J Struct. Funct. Genomics **2**: 37–51.

113 MÖLLER, S., D. R. CRONING AND R. APWEILER. 2001. Evaluation of methods for the prediction of membrane spanning regions. Bioinformatics **17**: 646–53.

114 MURVAI, J., K. VLAHOVICEK, C. SZEPESVARI AND S. PONGOR. 2001. Prediction of protein functional domains from sequences using artificial neural networks. Genome Res. **11**: 1410–7.

115 NADERI-MANESH, H., M. SADEGHI, S. ARAB AND A. A. MOOSAVI MOVAHEDI. 2001. Prediction of protein surface accessibility with information theory. Proteins **42**: 452–9.

116 NAIR, R. AND B. ROST. 2003. Better prediction of sub-cellular localization by combining evolutionary and structural information. Proteins **53**: 917–30.

117 NG, P. C., J. G. HENIKOFF AND S. HENIKOFF. 2000. PHAT: a transmembrane-specific substitution matrix. Predicted hydrophobic and transmembrane. Bioinformatics **16**: 760–6.

118 Norvell, J. C. and A. Z. Machalek. 2000. Structural genomics programs at the US National Institute of General Medical Sciences. Nat. Struct. Biol. **7** (**Suppl.**): 931.

119 Olmea, O., B. Rost and A. Valencia. 1999. Effective use of sequence correlation and conservation in fold recognition. J. Mol. Biol. **293**: 1221–39.

120 Olmea, O. and A. Valencia. 1997. Improving contact predictions by the combination of correlated mutations and other sources of sequence information. Fold. Des. **2**: S25–32.

121 Ortiz, A. R., A. Kolinski, P. Rotkiewicz, B. Ilkowski and J. Skolnick. 1999. Ab initio folding of proteins using restraints derived from evolutionary information. Proteins **Suppl. 3**: 177–85.

122 Ota, M., T. Kawabata, A. R. Kinjo and K. Nishikawa. 1999. Cooperative approach for the protein fold recognition. Proteins **37**: 126–32.

123 Ouali, M. and R. D. King. 2000. Cascaded multiple classifiers for secondary structure prediction. Protein Sci. **9**: 1162–76.

124 Panchenko, A., A. Marchler-Bauer and S. H. Bryant. 1999. Threading with explicit models for evolutionary conservation of structure and sequence. Proteins **Suppl. 3**: 133–40.

125 Park, J., K. Karplus, C. Barrett, R. Hughey, D. Haussler, T. Hubbard and C. Chothia. 1998. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. J. Mol. Biol. **284**: 1201–10.

126 Pauling, L. and R. B. Corey. 1951. Configurations of polypeptide chains with favored orientations around single bonds: two new pleated sheets. Proc. Natl Acad. Sci. USA **37**: 729–40.

127 Perutz, M. F., M. G. Rossmann, A. F. Cullis, G. Muirhead, G. Will and A. T. North. 1960. Structure of haemoglobin: a three-dimensional Fourier synthesis at 5.5 Å resolution, obtained by X-ray analysis. Nature **185**: 416–22.

128 Petersen, T. N., C. Lundegaard, M. Nielsen, H. Bohr, J. Bohr, S. Brunak, G. P. Gippert and O. Lund. 2000. Prediction of protein secondary structure at 80% accuracy. Proteins **41**: 17–20.

129 Pollastri, G. and P. Baldi. 2002. Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners. Bioinformatics **18** (**Suppl. 1**): S62–70.

130 Pollastri, G., P. Baldi, P. Fariselli and R. Casadio. 2001. Improved prediction of the number of residue contacts in proteins by recurrent neural networks. Bioinformatics **17**: S234–42.

131 Pollastri, G., P. Baldi, P. Fariselli and R. Casadio. 2002. Prediction of coordination number and relative solvent accessibility in proteins. Proteins **47**: 142–53.

132 Pollastri, G. and A. McLysaght. 2005. Porter: a new, accurate server for protein secondary structure prediction. Bioinformatics **21**: 1719–20.

133 Pollastri, G., D. Przybylski, B. Rost and P. Baldi. 2002. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. Proteins **47**: 228–35.

134 Prilusky, J., C. E. Felder, T. Zeev-Ben-Mordehai, E. Rydberg, O. Man, J. S. Beckmann, I. Silman and J. L. Sussman. 2005. FoldIndex©: a simple tool to predict whether a given protein sequence is intrinsically unfolded. Bioinformatics **21**: 3435–8.

135 Promponas, V. J., A. J. Enright, S. Tsoka, D. P. Kreil, C. Leroy, S. Hamodrakas, C. Sander and C. A. Ouzounis. 2000. CAST: an iterative algorithm for the complexity analysis of sequence tracts. Complexity analysis of sequence tracts. Bioinformatics **16**: 915–22.

136 Prusiner, S. B., M. R. Scott, S. J. DeArmond and F. E. Cohen. 1998. Prion protein biology. Cell **93**: 337–48.

137 Przybylski, D. and B. Rost. 2002. Alignments grow, secondary structure prediction improves. Proteins **46**: 195–205.

138 PRZYBYLSKI, D. AND B. ROST. 2004. Improving fold recognition without folds. J. Mol. Biol. **341**: 255–69.

139 PRZYTYCKA, T., R. AURORA AND G. D. ROSE. 1999. A protein taxonomy based on secondary structure. Nat. Struct. Biol. **6**: 672–82.

140 PTITSYN, O. B. AND A. V. FINKELSTEIN. 1983. Theory of protein secondary structure and algorithm of its prediction. Biopolymers **22**: 15–25.

141 PUNTA, M. AND B. ROST. 2005. PROFcon: novel prediction of long-range contacts. Bioinformatics **21**: 2960–8.

142 PUNTA, M. AND B. ROST. 2005. Protein folding rates estimated from contact predictions. J. Mol. Biol. **348**: 507–12.

143 RICHARDS, F. M. AND C. E. KUNDROT. 1988. Identification of structural motifs from protein coordinate data: secondary structure and first-level supersecondary structure. Proteins **3**: 71–84.

144 RICHARDSON, C. J. AND D. J. BARLOW. 1999. The bottom line for prediction of residue solvent accessibility. Protein Eng. **12**: 1051–4.

145 RIGDEN, D. J. 2002. Use of covariance analysis for the prediction of structural domain boundaries from multiple protein sequence alignments. Protein Eng. **15**: 65–77.

146 RIIS, S. K. AND A. KROGH. 1996. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. J. Comput. Biol. **3**: 163–83.

147 ROMERO, P., Z. OBRADOVIC, C. R. KISSINGER, J. E. VILLAFRANCA AND A. K. DUNKER. 1997. Identifying disordered regions in proteins from amino acid sequence. In Proc. IEEE Int. Conf. on Neural Networks, Houston, TX. Volume 1: 90–5.

148 ROMERO, P., Z. OBRADOVIC, C. R. KISSINGER, J. E. VILLAFRANCA, E. GARNER, S. GUILLIOT AND A. K. DUNKER. 1998. Thousands of proteins likely to have long disordered regions. Pac. Symp. Biocomput.: 437–48.

149 ROST, B. 2005. How to use protein 1D structure predicted by PROFphd. In WALKER, J. E. (ed.), *The Proteomics Protocols Handbook*. Humana, Totowa NJ: 875–901.

150 ROST, B. 1998. Marrying structure and genomics. Structure **6**: 259–63.

151 ROST, B. 1996. PHD: predicting one-dimensional protein structure by profile based neural networks. Methods Enzymol. **266**: 525–39.

152 ROST, B. 1995. TOBITS: threading one-dimensional predictions into three-dimensional structures. Proc. Int. Conf. Intell. Syst. Mol. Biol. **3**: 314–21.

153 ROST, B., R. CASADIO AND P. FARISELLI. 1996. Refining neural network predictions for helical transmembrane proteins by dynamic programming. Proc. Int. Conf. Intell. Syst. Mol. Biol. **4**: 192–200.

154 ROST, B. AND C. SANDER. 1994. Combining evolutionary information and neural networks to predict protein secondary structure. Proteins **19**: 55–72.

155 ROST, B. AND C. SANDER. 1994. Conservation and prediction of solvent accessibility in protein families. Proteins **20**: 216–26.

156 ROST, B. AND C. SANDER. 1993. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. Proc. Natl Acad. Sci. USA **90**: 7558–62.

157 ROST, B. AND C. SANDER. 1993. Prediction of protein secondary structure at better than 70% accuracy. J. Mol. Biol. **232**: 584–99.

158 ROST, B. AND C. SANDER. 2000. Third generation prediction of secondary structures. Methods Mol. Biol. **143**: 71–95.

159 ROST, B., C. SANDER AND R. SCHNEIDER. 1994. PHD – an automatic server for protein secondary structure prediction. CABIOS **10**: 53–60.

160 ROST, B., C. SANDER AND R. SCHNEIDER. 1994. Redefining the goals of protein secondary structure prediction. J. Mol. Biol. **235**: 13–26.

161 ROST, B. AND J. LIU. 2003. The PredictProtein server. Nucl. Acids Res. **31**: 3300–4.

162 RUSSELL, R. B., R. R. COPLEY AND G. J. BARTON. 1996. Protein fold recognition by mapping predicted secondary structures. J. Mol. Biol. **259**: 349–65.

163 SALAMOV, A. A. AND V. V. SOLOVYEV. 1995. Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignment. J. Mol. Biol. **247**: 11–5.

164 SAMUDRALA, R., E. S. HUANG, P. KOEHL AND M. LEVITT. 2000. Constructing side chains on near-native main chains for *ab initio* protein structure prediction. Protein Eng. **13**: 453–57.

165 SAMUDRALA, R., Y. XIA, E. HUANG AND M. LEVITT. 1999. *Ab initio* protein structure prediction using a combined hierarchical approach. Proteins **Suppl. 3**: 194–98.

166 SANDER, C. AND R. SCHNEIDER. 1991. Database of homology-derived structures and the structural meaning of sequence alignment. Proteins **9**: 56–68.

167 SAQI, M. 1995. An analysis of structural instances of low complexity sequence segments. Protein Eng. **8**: 1069–73.

168 SCHLESSINGER, A. AND B. ROST. 2005. Protein flexibility and rigidity predicted from sequence. Proteins **61**: 115–26.

169 SCHNEIDER, R. 1989. Sekundärstrukturvorhersage von Proteinen unter Berücksichtigung von Tertiärstrukturaspekten. *Diploma Thesis*. Department of Biology, University of Heidelberg.

170 SELBIG, J., T. MEVISSEN AND T. LENGAUER. 1999. Decision tree-based formation of consensus protein secondary structure prediction. Bioinformatics **15**: 1039–46.

171 SEN, T. Z., R. L. JERNIGAN, J. GARNIER AND A. KLOCZKOWSKI. 2005. GOR V server for protein secondary structure prediction. Bioinformatics **21**: 2787–8.

172 SHAO, Y. AND C. BYSTROFF. 2003. Predicting interresidue contacts using templates and pathways. Proteins **53**: 497–502.

173 SKLENAR, H., C. ETCHEBEST AND R. LAVERY. 1989. Describing protein structure: a general algorithm yielding complete helicoidal parameters and a unique overall axis. Proteins **6**: 46–60.

174 SKOLNICK, J., Y. ZHANG, A. K. ARAKAKI, A. KOLINSKI, M. BONIECKI, A. SZILAGYI AND D. KIHARA. 2003. TOUCHSTONE: a unified approach to protein structure prediction. Proteins **53 (Suppl. 6)**: 469–79.

175 SOLOVYEV, V. V. AND A. A. SALAMOV. 1994. Predicting α-helix and β-strand segments of globular proteins. Comput. Appl. Biol. Sci. **10**: 661–9.

176 SZENT-GYÖRGYI, A. G. AND C. COHEN. 1957. Role of proline in polypeptide chain configuration of proteins. Science **126**: 697.

177 THOMPSON, M. J. AND R. A. GOLDSTEIN. 1996. Predicting solvent accessibility: higher accuracy using Bayesian statistics and optimized residue substitution classes. Proteins **25**: 38–47.

178 UVERSKY, V. N., J. R. GILLESPIE AND A. L. FINK. 2000. Why are "natively unfolded" proteins unstructured under physiologic conditions? Proteins **41**: 415–27.

179 VON HEIJNE, G. 1992. Membrane protein structure prediction. J. Mol. Biol. **225**: 487–94.

180 VUCETIC, S., Z. OBRADOVIC, V. VACIC, *et al.* 2005. DisProt: a database of protein disorder. Bioinformatics **21**: 137–40.

181 WARD, J. J., L. J. MCGUFFIN, B. F. BUXTON AND D. T. JONES. 2003. Secondary structure prediction with support vector machines. Bioinformatics **19**: 1650–5.

182 WARD, J. J., J. S. SODHI, L. J. MCGUFFIN, B. F. BUXTON AND D. T. JONES. 2004. Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. J. Mol. Biol. **337**: 635–45.

183 WHEELAN, S. J., A. MARCHLER-BAUER AND S. H. BRYANT. 2000. Domain size distributions can predict domain boundaries. Bioinformatics **16**: 613–8.

184 WHISSTOCK, J. C. AND A. M. LESK. 2003. Prediction of protein function from protein sequence and structure. Q. Rev. Biophys. **36**: 307–40.

185 WOOTTON, J. C. AND S. FEDERHEN. 1996. Analysis of compositionally biased regions in sequence databases. Methods Enzymol. **266**: 554–71.

186 YUAN, Z., T. L. BAILEY AND R. D. TEASDALE. 2005. Prediction of protein B-factor profiles. Proteins **58**: 905–12.

187 ZEMLA, A., C. VENCLOVAS, K. FIDELIS AND B. ROST. 1999. A modified definition of *SOV*, a segment-based measure for protein secondary structure prediction assessment. Proteins **34**: 220–3.

188 ZHAO, Y. AND G. KARYPIS. 2003. Clustering in life sciences. Methods Mol. Biol. **224**: 183–218.

189 ZVELEBIL, M. J., G. J. BARTON, W. R. TAYLOR AND M. J. E. STERNBERG. 1987. Prediction of protein secondary structure and active sites using alignment of homologous sequences. J. Mol. Biol. **195**: 957–61.

10
Homology Modeling in Biology and Medicine

Roland L. Dunbrack, Jr.

1 Introduction

1.1 The Concept of Homology Modeling

To understand basic biological processes such as cell division, cellular communication, metabolism and development, knowledge of the three-dimensional (3-D) structure of the active components is crucial. Proteins form the key players in all of these processes, and the study of their diverse and elegant designs is a mainstay of modern biology. The Protein Databank (PDB) of experimentally determined protein structures [14] now contains nearly 40 000 entries, which can be grouped into about 1500 superfamilies [5]. The fact that proteins that share very little or no sequence similarity can have quite similar structures has led to the hypothesis that there are in fact only a few thousand different superfamilies [46, 84, 233] which have been adapted by a process of duplication, mutation and natural selection to perform all the biological functions that proteins accomplish.

Since it was first recognized that proteins can share similar structures [156], computational methods have been developed to build models of proteins of unknown structure based on related proteins of known structure [24]. Most such modeling efforts, referred to as homology modeling or comparative modeling, follow a basic protocol laid out by Greer [72, 73]: (i) identify a template structure related to the target sequence of unknown structure, and align the target sequence to the template sequence and structure; (ii) for core secondary structures and all well-conserved parts of the alignment, borrow the backbone coordinates of the template according to the sequence alignment of the target and template; (iii) for segments of the target sequence for which coordinates cannot be borrowed from the template because of insertions and deletions in the alignment (usually in loop regions of the protein) or because of missing coordinates in the template, build these segments using some construction method based on our knowledge of the determinants of protein structure; (iv) build side chains determined by the target sequence on to the

Bioinformatics – From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

backbone model built from the template structure and loop construction; (v) refinement of the model from the template backbone and toward the target structure.

The alignment step may involve a number of different strategies, including manual adjustment, even after the template structure or structures have been identified. Steps (iii) and (iv), backbone and side chain modeling, may be coupled, since certain backbone conformations may be unable to accommodate the required side-chains in any low-energy conformation. The refinement step involves moving beyond the aligned part of the backbone fixed in the template position and instead allowing it to adjust to the new sequence. For instance, two helices packed against each other may move apart to accommodate larger side-chains.

An alternative strategy has been developed by Blundell and colleagues, based on averaging a number of template structures, if these exist, rather than using a single structure [18,207,208]. More complex procedures based on reconstructing structures (rather than perturbing a starting structure) by satisfying spatial restraints using distance geometry [78] or molecular dynamics and energy minimization [118,173,174,180] have also been developed.

Many methods have been proposed to perform each of the steps in the homology modeling process. There are also a number of research groups that have developed complete packages that take as input a sequence alignment or even just a sequence and develop a complete model. In this chapter, we describe some of the basic ideas that drive loop and side-chain modeling individually as well as the complete modeling process. This chapter is a revised version of one that was published in 2001 [48]. In this revision, we emphasize those methods for which *usable* programs are currently publicly available. We also discuss more extensively the concept of modeling from the biological unit, including complexes of proteins with other proteins, DNA and ligands. The identification and alignment steps are covered in Chapters 3 and 11.

1.2 How do Homologous Protein Arise?

By definition, homologous proteins arise by evolution from a common ancestor. However, there are several different mechanisms for this and these are illustrated in Figure 1. The first is random mutation of individual nucleotides that change protein sequence, including missense mutations (changing the identity of a single amino acid) as well as insertions and deletions of a number of nucleotides that result in insertion and deletion of amino acids. As a single species diverges into two species, a gene in the parent species will continue to exist in the divergent species and over time will gather mutations that change the protein sequence. In this case, the genes in the different organisms will

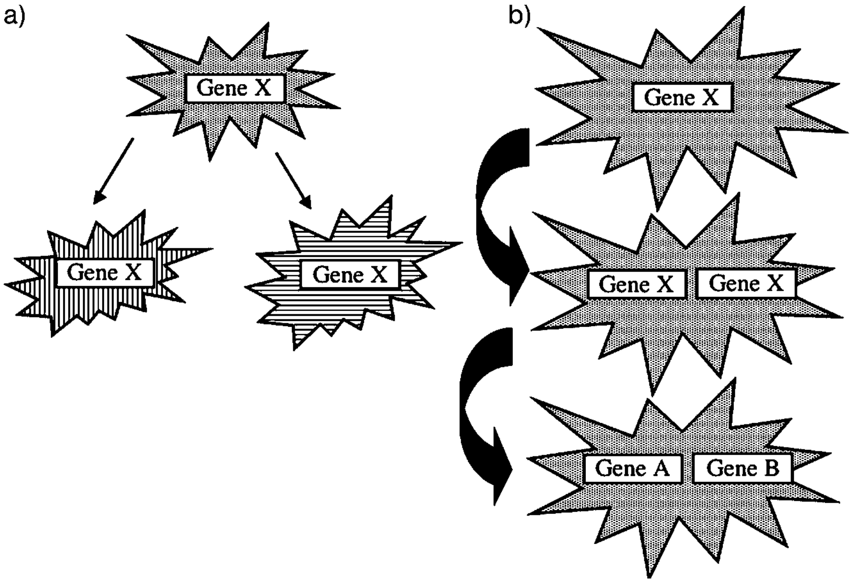


Figure 1 Orthologs versus paralogs. Schematic of the evolutionary process that gives rise to homologous proteins. (a) A single gene X in one species is retained as the species diverges into two separate species. The genes in these two species are *orthologous*. (b) A single gene X in one species is duplicated. As each gene gathers mutations, it may begin to perform new functions, or the two genes may specialize in carrying out two or more functions of the ancestral gene, thus improving the fitness of the organism. These genes in one species are paralogous. If the species diverges, each daughter species may maintain the duplicated genes, and therefore each species contains an ortholog and a paralog to each gene in the other species.

usually maintain the same function. These genes are referred to as orthologs of one another. A second mechanism is duplication of a gene or of a gene segment within a single organism or germ line cell. As time goes by, the two copies of the gene may begin to gather mutations. If the template gene performed more than one function, e.g. similar catalytic activity on two different substrates, one of the duplicated genes may gain specificity for one of the reactions, while the other gene gains specific activity for the other. If this divergence of specificity in the two proteins is advantageous, the duplication will become fixed in the population. These two genes are paralogs of one another. If the species with the pair of paralogs diverges into two species, each species will contain the two paralogs. Each gene in each species will now have an ortholog and a paralog in the other species.

1.3 The Purposes of Homology Modeling

Homology modeling of proteins has been of great value in interpreting the relationships of sequence, structure and function. In particular, orthologous proteins usually show a pattern of conserved residues that can be interpreted in terms of 3-D models of the proteins. Conserved residues often form a contiguous active site or interaction surface of the protein, even if they are distant from each other in the sequence. With a structural model, a multiple alignment of orthologous proteins can be interpreted in terms of the constraints of natural selection and the requirements for protein folding, stability, dynamics and function.

For paralogous proteins, 3-D models can be used to interpret the similarities and differences in the sequences in terms of the related structure, but different functions of the proteins concerned [121]. In many cases, there are significant insertions and deletions and amino acid changes in the active or binding site between paralogs. However, by grouping a set of related proteins into individual families, orthologous within each group, the evolutionary process that changed the function of the ancestral sequences can be observed. Indeed, homology models can serve to help us identify which protein belongs to which functional group by the conservation of important residues in the active or binding site [62]. A number of recent papers have been published that use comparative modeling to predict or establish protein function [95,106,142,222, 225]; see also Chapter 33.

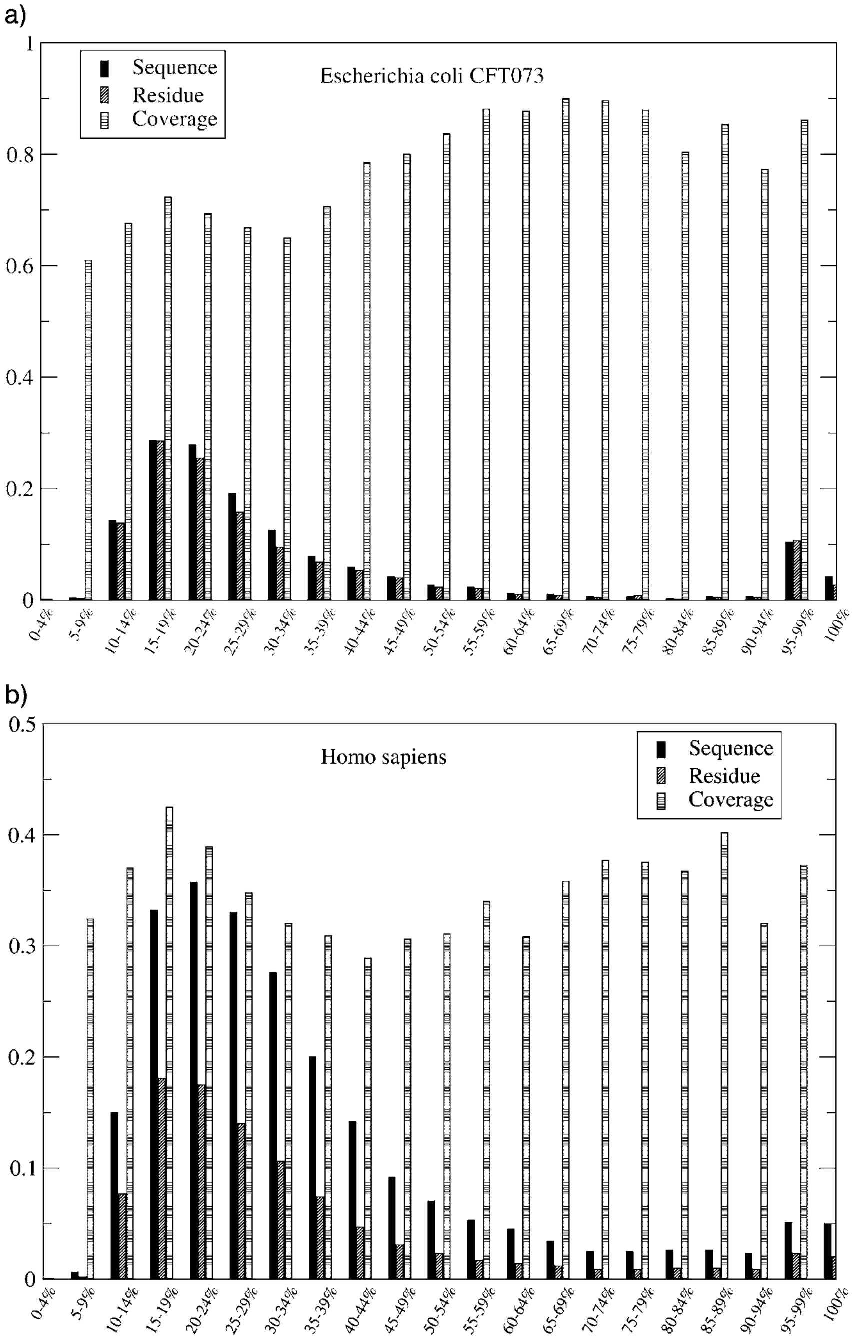
Another important use of homology modeling is to interpret point mutations in protein sequences that arise either by natural processes or by experimental manipulation. The human genome project has produced significant amounts of data concerning polymorphisms and other mutations potentially related to differences in susceptibility, prognosis and treatment of human disease. There are now many such examples, including the Factor V/Leiden R506Q mutation [247] that causes increased occurrence of thrombosis, mutations in cystathionine β-synthase that cause increased levels of homocysteine in the blood, a risk factor for heart disease [101], and BRCA1 for which many sequence differences are known, some of which may lead to breast cancer [34]. At the same time, there are many polymorphisms in important genes that have no discernible effect on those who carry them. At least for some of these, there may be some effect that has yet to be measured in a large enough population of patients and therefore the risk of cancer, heart disease or other illness to these patients is unknown. This is yet another important application of homology modeling, since a good model may indicate readily which mutations pose a likely risk and which do not [92].

Homology models may also be used in computer-aided drug design, especially when a good template structure is available for the target sequence. For enzymes that maintain the same catalytic activity, the active site may be sufficiently conserved such that a model of the protein provides a reasonable target for computer programs which can suggest the most likely compounds that will bind to the active site (see also Chapter 16). This has been used successfully in the early development of HIV protease inhibitors [223, 224] and in the development of anti-malarial compounds that target the cysteine protease of *Plasmodium falciparum* [166].

1.4 The Effect of the Genome Projects

The many genome projects now completed or underway have greatly affected the practice of homology modeling of protein structures. First, the many new sequences have provided a large number of targets for modeling. Second, the large amount of sequence data makes it easier to establish remote sequence relationships between proteins of unknown structure and those of known structure on which a model can be built. The most commonly used methods for establishing sequence relationships such as PSI-BLAST [3] are dependent on aligning many related sequences to compile a pattern or profile of sequence variation and conservation for a sequence family. This profile can be used to search among the sequences in the PDB for a relative of the target sequence (see Chapter 11). The more numerous and more varied sequences there are in the family, the more remote are the homologous relationships that can be determined and the more likely it is that a homologous template for a target sequence can be found. Third, it is likely that the accuracy of sequence alignments between the sequence of unknown structure that we are interested in and the protein sequence of a template are also greatly improved with profiles established from many family members of the target sequence [184]. Fourth, the completion of a number of microbial genomes has prodded a similar effort among structural biologists to determine the structures of representatives of all common protein sequence families, or all proteins in a prototypical genome, such as *Mycobacterium tuberculosis* [15, 126, 163, 210, 241]. Protein structures determined by X-ray crystallography or NMR spectroscopy are being solved at a much faster pace than was possible even 10 years ago. The great increase in the number of solved protein structures has a great impact on the field of homology modeling, since it becomes ever more likely that there will be a template structure in the PDB for any target sequence of interest [221] (see also Chapter 13).

Given the current sequence and structure databases, it is of interest to determine what fraction of sequences might be modeled and the range of sequence identities between target sequences and sequences of known structure. In Figure 2, we show histograms of sequence identities of the sequences in several genomes and their nearest relatives of known structure in the PDB. These relationships were determined with PSI-BLAST as described in the legend. PSI-BLAST is fairly sensitive in determining distant homology relationships [85, 184, 232], although more sensitive techniques exist (see Chapter 11). The results indicate that on average 30–40% of genomic protein sequences are


a)
Escherichia coli CFT073
▮ Sequence
▨ Residue
▤ Coverage
1
0.8
0.6
0.4
0.2
0
0-4%
5-9%
10-14%
15-19%
20-24%
25-29%
30-34%
35-39%
40-44%
45-49%
50-54%
55-59%
60-64%
65-69%
70-74%
75-79%
80-84%
85-89%
90-94%
95-99%
100%
b)
Homo sapiens
▮ Sequence
▨ Residue
▤ Coverage
0.5
0.4
0.3
0.2
0.1
0
0-4%
5-9%
10-14%
15-19%
20-24%
25-29%
30-34%
35-39%
40-44%
45-49%
50-54%
55-59%
60-64%
65-69%
70-74%
75-79%
80-84%
85-89%
90-94%
95-99%
100%

easily identified as related to proteins of known structure, which presents a large number of potential targets for homology modeling. However, it should also be pointed out that the average sequence identity between target sequences and template structures in the PDB is less than 25%.

The low sequence identity between target and template sequences in Figure 2 presents a major challenge for homology modeling practitioners, since a major determinant in the accuracy of homology modeling is the sequence identity between the target sequence and the sequence of the template structure. At levels below 30% sequence identity, related protein structures diverge significantly and there may be many insertions and deletions in the sequence [31]. At 20% sequence identity, the average RMSD of core backbone atoms is 2.4 Å [31, 169]. However, as demonstrated in Figure 2, it is likely that we will most often face a situation where the target and template sequences are remotely related. Most widely used homology modeling methods have been predicated on much higher sequence identities between template and target, usually well above 30% [43,155,181]. What methods should be used at sequence identities in the 10–30% range is of crucial importance in this post-genomic era.

2 Input Data

To produce a protein model that will be useful and informative requires more than placing a new sequence onto an existing structure. A large amount of sequence data and other kinds of experimental data can often be gathered on the target sequence and on its homolog of known structure to be used for model building. This information can be used to build a better model *and* as the data to be interpreted in light of the model. The goal is to forge an

←

Figure 2 Distribution of sequence identities between protein in four genomes and their closest homologs in the PDB for those sequences in genomes with homologs in the PDB. PSI-BLAST was used to search the nonredundant protein sequence database with a representative set of PDB sequences as queries. The program was run for four iterations, with a maximum E-value of 0.0001 used to determine sequences which are included in the position-specific similarity matrix. After four iterations, each matrix was used to search each of the four genomes. Coiled-coil and low-sequence complexity sequences were removed from each genome and the nonredundant sequence database. All hits in the genomes with E-values less than 0.001 were saved and the histograms were built from the PSI-BLAST-derived sequence identities.

integrated model of the protein sequence, structure, and function, not merely to build a structure. In Table 1, we list the kinds of information that might be available for a target protein and how these data might be processed.

Table 1 Input information for homology model building

Target sequence
• Target orthologous relatives (from PSI-BLAST)
• Target paralogous relatives (from PSI-BLAST)
• Multiple sequence alignment of orthologs and paralogs (either BLAST multiple alignment or (preferably) other multiple alignment program)
• Sequence profile of ortho/paralogs
Template sequences and structures
• Homolog(s) of known structure [template(s)] determined by database search methods (BLAST, PSI-BLAST, intermediate sequence search methods, HMMs, fold recognition methods)
• Template orthologous sequences
• Template paralogous sequences
• Multiple sequence alignment of template orthologs and paralogs
• Biological units of available templates from RCSB and EBI/PQS
Alignment of target sequence to template sequence and structure
• Pairwise alignment
• Profile alignment
• Multiple sequence alignment of target and template sequence relatives
• Profile–profile alignment
• Fold recognition alignment
• Visual examination of proposed alignments and manual adjustment
• Assessment of confidence in alignment by residue (some regions will be more conserved than others)
Structure alignment of multiple templates, if available
• Align by structure (fssp, VAST, CE, etc.)
• Compare sequence alignments from structure to sequence alignments from multiple sequence alignments (see above)
Experimental information
• Mutation data (site directed, random, naturally occurring)
• Functional data, e.g. DNA binding, ligands, metals, catalysis, etc.
• Oligomer data, e.g. analytical ultracentrifugation, native gel electrophoresis

Since proteins act through their interactions with other molecules, it is important to gather information on known or putative ligands or binding partners of the target. Indeed, the target of the modeling may not be a single protein but a protein complex. As the number of structures of multiprotein complexes increases, there are more and more templates for this kind of modeling. Many proteins act as homo-multimers and so it is important to know whether the goal of modeling is a dimer or tetramer or other multimer of the target. While this may not be known for the modeling target itself, it may be known experimentally for homologs of the target through various experiments, including analytical ultracentrifugation, native gel electrophoresis and of course X-ray crystallography (see below in this section). Information on protein–protein interactions of the target, DNA binding, and other ligands such as ions and organic substrates or cofactors is also important and may be included in the modeling.

With the large amount of sequence information available, it is almost always possible to produce a multiple alignment of sequences related to the target protein. The first step in modeling therefore is to use a database search program such as PSI-BLAST [3] against a nonredundant protein sequence database such as NCBI's *nr* database [13] or the curated UniProt database [7]. With some care, a list of relatives to the target sequence can be gathered and aligned. PSI-BLAST provides reasonable multiple alignments, but it may be desirable to take the sequences identified by the database search and realign them with a multiple sequence alignment program such as ClustalW [211] and Muscle [55]. PSI-BLAST tends to create multiple sequence alignments with many gaps, because insertions relative to the query may be placed at slightly different positions (see also Chapter 3).

It may be that a database search consisting of several rounds of PSI-BLAST will provide one or more sequences of known 3-D structure. If this is not the case then more sensitive methods based on fold recognition or hidden Markov models (HMMs) [6, 8, 23, 53, 54, 93] of protein superfamilies may identify a suitable template structure (see Chapter 11). Once a template structure is identified, a sequence database search will provide a list of relatives of the template, analogous to searches for relatives of the target. At this stage it is useful to divide the sequences related to the target into orthologs of either the target or the template (or both). The sequence variation within the set of proteins that are orthologous to the target provides information as to what parts of the sequence are most conserved and therefore likely to be most important in the model. Similarly variation in the set of proteins that are orthologous to the template provide a view of the template protein family that can be used to identify features in common or distinct in the template and target families. These features can be used to evaluate and adjust a joint multiple alignment of both families.

If there are multiple structures in the PDB that are homologous to the target sequence, then it is necessary to evaluate them to determine which PDB entry will provide the best template structure and whether it will be useful to use more than one structure in the modeling process. In the case of a single sequence that occurs in multiple PDB entries, it is usually a matter of selecting the entry with the highest resolution or the most appropriate ligands (DNA, enzyme inhibitors, metal ions). In other cases, there may be more than one homolog related to the target sequence, and the task is to select the one more closely related to the target or to combine information from more than one template structure to build the model. To do this, a structure alignment of the potential templates can be performed with one of a number of available computer programs (Dali [82], CE [194], etc.). From alignments of the target to the available templates, the location of insertions and deletions can be observed, and often it will be clear that one template is better than others. This may not be uniform, however, such that some regions of the target may have no insertions or deletions with respect to one template, but other regions are more easily aligned with the other template. In this case, a hybrid structure may be constructed [207].

As noted above, it may be desirable to build a particular multimer of the target sequence. It is therefore important to gather information on the biological units for the available template structures. The biological unit is defined as the likely oligomeric state of a protein in its relevant biological context. By contrast, the asymmetric unit is the object for which there is independent experimental information in the crystallographic experiment. The asymmetric unit may be a monomer or dimer or higher multimer of the protein or proteins in the crystal. Quite often the biological unit is present within the crystal and may or may not coincide with the asymmetric unit. In some cases it may be made of parts or all of more than one asymmetric unit. In other cases, the asymmetric unit is composed of more than one biological unit.

The possibilities are illustrated in Figure 3, where the asymmetric units from three different crystal structures of hemoglobin are shown. Hemoglobin is a tetramer consisting of two α- and two β-chains. In the first structure, the asymmetric unit consists of an entire tetramer and therefore coincides with the biological unit. The second structure contains only an α–β dimer and therefore the biological unit is constructed with the space group symmetry operators to form a tetramer. In the third case, the biological unit consists of two tetramers and therefore contains two copies of the biological unit.

The probable biological units are obtainable from both the PDB and the European Bioinformatics Institute (EBI) from their Protein Quaternary Server (PQS) [80]. Often these two sources do not agree on the biological unit for a particular PDB entry and they should be interpreted as hypothetical

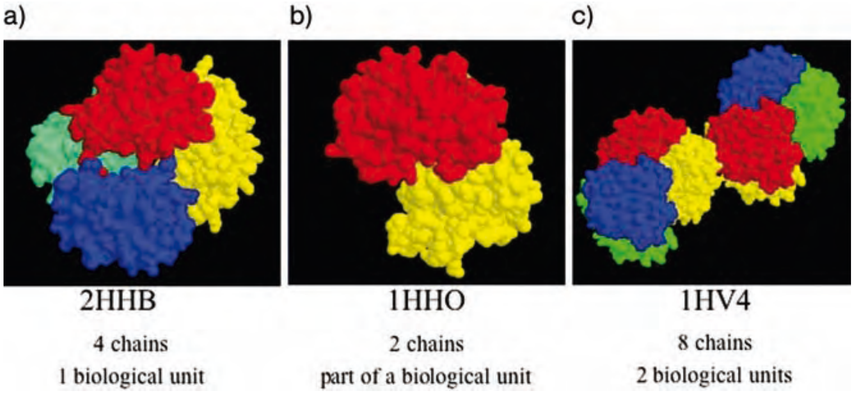


Figure 3 Asymmetric units for hemoglobin from three different structures. The biological unit consists of four chains (two α- and two β-chains). Three scenarios are shown: (a) the asymmetric unit consists of exactly one biological unit, (b) the asymmetric unit is smaller than a biological unit (in this case, it is one half of a biological unit) and (c) the asymmetric unit is larger than the biological unit (in this case, it is two biological units).

oligomers. By comparing the asymmetric units with those from the PDB and the EBI, we found that for over 50% of structures, the asymmetric unit does not correspond to the biological unit for PDB or PQS or both. The PDB and PQS agree 80% of the time on the biological unit (see Section 4.3). It is therefore important to choose a template that has the correct multimer status in its biological unit and to use this biological unit in the modeling process, rather than the asymmetric unit.

Finally, any other experimental data available on the target or template proteins may be very helpful in producing and interpreting a structural model. These can include inhibitor studies, DNA binding and sequence motifs, proteolysis sites, metal binding, mutagenesis data, etc. A number of databases are available on the web that summarize information on particular genes or that collect information on mutations and polymorphisms linked to disease, including: the Cancer Genome Anatomy Project [201], the Online Mendelian Inheritance in Man (OMIM) [76] and the Human Gene Mutation Database [102, 200].

3 Methods

3.1 Modeling at Different Levels of Complexity

Once an alignment is obtained between the target and a protein of known structure (as described in Section 2, and in Chapters 3 and 10), it is possible to build a series of models of increasing sophistication.

(i) *Simple model*: keep backbone and conserved side chains by renaming and renumbering coordinates in the template structure with the new sequence using the alignment of target and template; rebuild other side chains using a side chain modeling program (e.g. SCWRL [22, 30, 47]); do not model insertions or deletions (i.e. do not build new loops and do not close up gaps).

(ii) *Stepwise model*: borrow core backbone from template structure, minus coil regions with insertions or deletions in the sequence alignment; rebuild core side chains; rebuild coil regions with loop prediction method in conjunction with side-chain prediction method. Core backbone and side chains may or may not be held fixed during loop prediction. The entire model may be refined using energy minimization, Monte Carlo or molecular dynamics techniques.

(iii) *Jigsaw model*: borrow backbone from a common core of several structurally aligned templates, using loop regions from different templates according to the alignments, usually keeping those loops for which there is no gap in the alignment with the target sequence. Some loops may need to be modeled.

(iv) *Global model*: build entire protein from spatial restraints drawn from known structure(s) and sequence alignment (e.g. MODELLER [174,180]).

It is not always the case that more sophisticated models are better than simpler, less-complete ones. If elements of secondary structures are allowed to move away from their positions in the template and large changes are made to accommodate insertions and deletions, it may be the case that the model is further away from the target structure (if it were known) than the template structure was to begin with. This is the "added value" problem discussed by John Moult at the Critical Assessment of Protein Structure Prediction (CASP) meetings [138,140,141]. We would like methods that move the template structure closer to the target structure, such that they "add value" to a simple model or unrefined stepwise model based on an unaltered template structure, with side-chains replaced. Extensive energy minimization or molecular dynamics simulations often bring a model further away from the correct structure than toward it [59, 98].

The simple model is sometimes justified when there are no insertions and deletions between the template and target or when these sequence length changes are far from the active site or binding site of the protein to be modeled. This often occurs in orthologous enzymes that are under strong selective pressure to maintain the geometry of the active site. Even in nonorthologous enzymes, sometimes we are most interested in an accurate prediction of the active-site geometry and not in regions of the protein distant from the active site.

A stepwise model is probably the most common method used in homology modeling, since it is conceptually simpler than the more complex models and since each piece can be constructed and examined in turn. Some programs therefore proceed by taking the sequence–structure alignment, removing all regions where there are insertions and deletions, and reconstructing loops and side-chains against the fixed template of the remaining atoms. Some methods may also allow all parts of the template structure to adjust to the changes in sequence and insertions and deletions. This usually takes the form of a Monte Carlo or molecular dynamics simulation [118]. A global model, as described above, rebuilds a structure according to constraints derived from the known template structure or structures. This is in contrast to stepwise models that proceed essentially by replacing parts of the template structure and perhaps perturbing the structure.

Many computer programs for homology modeling are developed to solve a single problem, such as loop or side-chain building, and may not be set up to allow all atoms of the protein to adjust or to model many components simultaneously. In many cases these methods have been tested by using simplified modeling situations. Such examples include experiments with removing and rebuilding loops onto single protein structures, and stripping and rebuilding all side chains. In the next sections we review some of the work in these two areas.

3.2 Side-chain Modeling

3.2.1 Input Information

Side-chain modeling is a crucial step in predicting protein structure by homology, since side-chain identities and conformations determine the specificity differences in enzyme active sites and protein binding sites. The problem has been described as "solved" [117], although new methods [120, 133, 157, 193, 234] or improvements on older ones [30] continue to be published. Some side-chain prediction methods stand on their own and are meant to be used with a fixed backbone conformation and sequence to be modeled given as input. Other methods have been developed in the context of general homology modeling methods, including the prediction of insertion-deletion regions. Even when using general modeling procedures, such as MODELLER, it may be worthwhile subsequently to apply a side-chain modeling step with other programs optimized for this purpose [220]. This is especially the case when side-chain conformations may be of great importance to interpretation of the model. It is also often the case that insertion-deletion regions are far away from the site of interest and loop modeling may be dispensed with. Indeed, significant alterations of the backbone of the template, if they are not closer to the target to be modeled (if it were known) than the template, may in fact result in poorer side-chain modeling than if no loop modeling were performed. As described above, the choice of template may depend not only on sequence identity but also on the absence of insertions and deletions near the site of interest. If this is successful, side-chain modeling rises in importance in relation to loop prediction.

Side-chain prediction methods described in detail in the literature have a long history although only a small number of programs are currently publicly available (see Table 2). Nearly all assume a fixed backbone, which may be from a homologous protein of the structure to be modeled, or may be the actual X-ray backbone coordinates of the protein to be modeled. Many methods have in fact only been tested by replacing side-chains onto backbones taken from the actual 3-D coordinates of the proteins being modeled ("self-backbone predictions"). Nevertheless, these methods can be used for homology modeling by first substituting the target sequence onto the template backbone and then modeling the side chains. When a protein is modeled from a known structure, information on the conformation of some side chains may be taken from the template [22, 30, 204]. This is most frequently the case when the template and target residue are identical, in which case the template residue's Cartesian coordinates may be used. These may be kept fixed as the other side chains are placed and optimized or they may be used only as a starting conformation and optimized with all other side chains. Only a small number of methods use information about nonidentical side chains borrowed from the template. For instance, Phe ↔ Tyr substitutions only require the building or removal of a hydroxyl group while Asn ↔ Asp substitutions require changing one of δ-atoms from NH_2 to O or vice versa. Summers and Karplus [203, 204] used a more detailed substitution scheme, by which for instance the χ_1 angle of very different side-chain types (e.g. Lys ↔ Phe) might be used in building side chains. In the long run, this is probably not advantageous, since the conformational preferences of nonsimilar side-chain types may be quite different from each other [50].

Table 2 Publicly available side-chain prediction programs

Program	Availability	Website
SMD	download	http://condor.urbb.jussieu.fr/Smd.php
Confmat, Decorate	web	http://lorentz.immstr.pasteur.fr/website/projects
CARA/GeneMine	download	http://www.bioinformatics.ucla.edu/genemine
RAMP	download	http://www.ram.org/computing/ramp
SCAP	download	http://honiglab.cpmc.columbia.edu/programs/sidechain
SCWRL	download	http://dunbrack.fccc.edu/scwrl
Maxsprout/Torso	web	http://www.ebi.ac.uk/maxsprout
SCATD	download	http://www.bioinformatics.uwaterloo.ca/%7Ej3xu
PLOP	download	http://francisco.compbio.ucsf.edu/~jacobson/plop_manual/plop_overview

3.2.2 Rotamers and Rotamer Libraries

Nearly all side-chain prediction methods depend on the concept of side-chain *rotamers* (reviewed in Ref. [49]). From conformational analysis of organic molecules, it was predicted long ago [182,183] that protein side chains should attain a limited number of conformations because of steric and dihedral strain within each side chain, and between the side chain and the backbone. Dihedral strain occurs because of Pauli exclusion between bonding molecular orbitals in eclipsed positions [94]. For sp^3–sp^3 hybridized bonds, the energy minima for the dihedral are at the staggered positions that minimize dihedral strain at approximately 60°, 180°, and –60°. For sp^3–sp^2 bonds, the minima are usually narrowly distributed around +90° or –90° for aromatics and widely distributed around 0° or 180° for carboxylates and amides (e.g. Asn/Asp χ_2 and Glu/Gln χ_3).

As crystal structures of proteins have been solved in increasing numbers, a variety of rotamer libraries have been compiled with increasing amounts of detail and greater statistical soundness, i.e. with more structures at higher resolution [12,17,50–52,88,125,132,161,187,212]. The earliest rotamer libraries were based on a small number of structures [12, 17, 88, 161]. Even the widely used Ponder and Richards library was based on only 19 structures, including only 16 methionines [161]. The most recent libraries are based on over 850 structures with resolution of 1.7 Å or better and mutual sequence identity less than 50% between any two chains used.

Most rotamer libraries are backbone-conformation-independent. In these libraries, the dihedral angles for side chains are averaged over all side chains of a given type and rotamer class, regardless of the local backbone conformation or secondary structure. The most recent of these is by Lovell and coworkers [125], who derived a more accurate backbone-independent rotamer library by eliminating side chains of low stereochemical quality, including those with high B-factors, steric conflicts in the presence of predicted hydrogen atom locations, and other factors. The statistical analysis does not rely on a parametric distribution function such as the normal model, and hence can model factors like skew in an unbiased way.

Several libraries have been proposed that are dependent on the conformation of the local backbone [50–52,132,187]. McGregor and coworkers [132] and Schrauber and coworkers [187] compiled rotamer probabilities and dihedral angle averages in different secondary structures We have used Bayesian statistical methods to compile a backbone-dependent rotamer library with rotamer probabilities and average angles and standard deviations at all values of the backbone dihedral angles ϕ and ψ in 10° increments [49–52]. The current version of this library is based on 850 chains with resolution better than 1.7 Å and less than 50% mutual sequence identity.

Finally, there is an alternative form of a rotamer library that includes large numbers of conformations of each side-chain type in the form of Cartesian coordinates. These libraries therefore include variation in bond lengths and bond angles, as well as dihedral angles. They are generally used for fine sampling of side-chain positions in the context of side-chain prediction. For instance, Xiang and Honig [234] produced a library consisting of 7560 conformations for use in their side-chain prediction method from a set of 297 high-resolution structures. The variation in bond angles and dihedrals away from average values is particularly useful for larger side chains for which a small change in an angle near the base of the side chain may cause large motions of atoms at the far end of the side chain. Other groups have also used large rotamer libraries to introduce flexibility about mean dihedral angles of rotamers as well as variation in bond lengths and bond angles [157, 193].

3.2.3 Side-chain Prediction Methods

Side-chain prediction methods can be classified in terms of how they treat side-chain dihedral angles (rotamer library, grid or continuous dihedral angle distribution), bond lengths and bond angles (fixed, variable, sampled from Cartesian conformers), potential energy function used to evaluate proposed conformations, and search strategy.

The potential energy functions in side-chain prediction methods have varied tremendously from simple steric exclusion terms to full molecular mechanics potentials. In most cases, the potential energy function is a standard Lennard–Jones potential:

$$E(r) = 4\varepsilon \left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right] . \tag{1}$$

In this equation, r is the distance between two nonbonded atoms, and ε and σ are parameters that determine the shape of the potential. This potential has a minimum at the distance $r = 2^{1/6}\sigma$ and a well depth of ε. Different values of σ and ε may be chosen for different pairs of atom types. Some potential energy functions for side chains may also include a hydrogen bond term. Depending on the potential parameters, these potentials may not accurately model the relative energies of rotamers for each side-chain type that are determined from local interactions within each side chain and between the side chain and the local backbone. For instance, in molecular mechanics potentials, interactions between atoms connected by three covalent bonds (atoms i and $i + 3$ in a chain) are not usually treated by van der Waals terms, but rather in torsion terms of the form [127]:

$$E(\tau) = \sum_m K_m \cos(m\tau + a_m) \tag{2}$$

where the sum over m may include 1-, 2-, 3-, 4- and 6-fold cosine terms. The K_m and a_m are constants specific for each dihedral angle and each term in the sum. These torsion terms are included in some side-chain prediction methods, but ignored in others [96].

Electrostatic interactions in the form of a Coulomb potential have been included in methods that rely on full molecular mechanics potentials, usually with a distance-dependent dielectric, $\varepsilon(r) = r$:

$$E = \frac{q_i q_j}{\varepsilon(r) r} \tag{3}$$

Solvent interactions are also usually ignored, since these can be difficult or expensive to model properly (for exceptions, see [120,185,228]).

A number of side-chain methods use an energy term based on the probability of rotamers as a function of backbone conformation. These probabilities are given in the backbone-dependent rotamer library, and the energy function is usually of the form:

$$E_i = -K \ln \left(\frac{p_i(\phi, \psi, R)}{p_{\max}(\phi, \psi, R)} \right) \tag{4}$$

where the energy of rotamer i is expressed as a function of the probability of this rotamer given the backbone dihedrals ϕ and ψ and the residue type R, and the probability of the most common rotamer for the same backbone dihedrals and residue type. The constant K is empirical and can be optimized given the other terms in the energy function.

Side-chain conformation prediction incurs the risk of combinatorial explosion, since there are on the order of n_{rot}^N possible conformations, where n_{rot} is the average number of rotamers per side chain and N is the number of side chains. However, in fact, the space of conformations is much smaller than that, since side chains can only interact with a small number of neighbors, and in most cases clusters of interacting side chains can be isolated and each cluster can be solved separately [22, 212]. Also, many rotamers have prohibitively large interactions with the backbone and are at the outset unlikely to be part of the final predicted conformation. These can be eliminated from the search early on.

Many standard search methods have been used in side-chain conformation prediction, including Monte Carlo simulation [83,109,116,120,137,167], simulated annealing [86], self-consistent mean field calculations [96,133,134], the dead-end elimination (DEE) method [40–42,70,107,123,159], neural networks [99] and graph theory [30,111,236].

Self-consistent mean field calculations represent each side chain as a set of conformations, each with its own probability. Each rotamer of each side chain has a certain probability, $p(r_i)$. The total energy is a weighted sum of the interactions with the backbone and interactions of side chains with each other:

$$E_{\text{tot}} = \sum_{i=1}^{N} \sum_{r_i=1}^{n_{\text{rot}}(i)} p(r_i) E_{bb}(r_i) + \sum_{i=1}^{N-1} \sum_{r_i=1}^{n_{\text{rot}}(i)} \sum_{j=i+1}^{N} \sum_{r_j=1}^{n_{\text{rot}}(j)} p(r_i) p(r_j) E_{\text{sc}}(r_i, r_j) \tag{5}$$

In this equation, $p(r_i)$ is the density or probability of rotamer r_i of residue i, $E_{bb}(r_i)$ is the energy of interaction of this rotamer with the backbone, and $E_{\text{sc}}(r_i,r_j)$ is the interaction energy (van der Waals, electrostatic) of rotamer r_i of residue i with rotamer r_j of residue j. Some initial probabilities are chosen for the ps in Eq. (5) and the energies calculated. New probabilities $p\prime(r_i)$ can then be calculated with a Boltzmann distribution based on the energies of each side chain and the probabilities of the previous step:

$$\begin{aligned} E(r_i) &= E_{bb}(r_i) + \sum_{j=1, j\neq i}^{N} \sum_{r_j=1}^{n_{\text{rot}}(j)} p(r_j) E_{\text{sc}}(r_i, r_j) \\ p'(r_i) &= \frac{\exp(-E(r_i)/kT)}{\sum_{r_i=1}^{n_{\text{rot}}(i)} \exp(-E(r_i)/kT)} \end{aligned} \tag{6}$$

Alternating steps of new energies and new probabilities can be calculated from the expressions in Eq. (6) until the changes in probabilities and energies in each step become smaller than some tolerance.

The DEE algorithm is a method for pruning the number of rotamers used in a combinatorial search by removing rotamers that cannot be part of the global minimum energy conformation [41, 42, 70, 107, 108, 123]. This method can be used for any search problem that can be expressed as a sum of single-residue terms and pairwise interactions. Goldstein's improvement on the original DEE can be expressed as follows [70]. If the total energy for all side chains is expressed as the sum of singlet and pairwise energies:

$$E = \sum_{i=1}^{N} E_{bb}(r_i) + \sum_{i=1}^{N-1} \sum_{j>i}^{N} E_{\text{sc}}(r_i, r_j) \tag{7}$$

then a rotamer r_i can be eliminated from the search if there is another rotamer s_i for the same side chain that satisfied the following equation:

$$E_{bb}(r_i) - E_{bb}(s_i) + \sum_{j=1, j\neq i}^{N} \min_{r_j} \left\{ E_{\text{sc}}(r_i, r_j) - E_{\text{sc}}(s_i, r_j) \right\} > 0 \tag{8}$$

In words, rotamer r_i of residue i can be eliminated from the search if another rotamer of residue i, s_i, always has a lower interaction energy with all other side chains regardless of which rotamer is chosen for the other side chains. More powerful versions have been developed that eliminate certain pairs of rotamers from the search [42,70,123]. DEE-based methods have also proved very useful in protein design, where there is variation of residue type as well as conformation at each position of the protein [37,71,218].

The current SCWRL algorithm [30] uses graph theory to solve the combinatorial problem. In this method, each side chain in the protein is considered a node in an undirected graph. An edge exists between two nodes *i* and *j* if at least one rotamer of residue *i* and one rotamer of residue *j* interact with each other, i.e. have a nonzero interaction energy. This produces a number of separate graphs that are not connected to each other. Each of these graphs can then be solved for the minimum energy conformation of the residues in the graph. To accomplish this, each separate graph is broken up into its biconnected components, as shown in Figure 4(a). Biconnected components are cycles or nested cycles or bridges consisting of two nodes connected by an edge. Two biconnected components share a node called an articulation point, which when removed from the graph breaks the graph into two (or more) connected subgraphs. The global minimum of the energy can be found by beginning on the outside of the graph with biconnected components that have only one articulation point. For each rotamer of the articulation point, the minimum energy of the other rotamers is found and stored with the rotamer of the articulation point. Then the biconnected graph is "collapsed" onto the articulation point residue. This residue now contains information on all the residues in the biconnected component. The procedure continues to collapse biconnected components, until a single component is left, as shown in Figure 4.

Recently, two papers [111,236] have appeared that extend the graph theory algorithm further so that the smallest groups that need to be searched are much smaller than biconnected components. In this method, some nodes can be removed from the graph by collapsing a node or nodes onto an *edge*. This is shown in Figure 4(b), in which a single node that has two neighbors in the graph is collapsed onto an edge between the two neighbors. The new energy of each rotamer pair for residues *i* and *j* is now:

$$E_{\text{pair}}^{\text{new}}(r_i, r_j) = E_{\text{pair}}^{\text{old}}(r_i, r_j) + \min_{r_k} \left\{ E_{\text{self}}(r_k) + E_{\text{pair}}(r_i, r_k) + E_{\text{pair}}(r_j, r_k) \right\} \quad (9)$$

The size of the smallest group that must be solved combinatorially is called the tree width and is related to the size of the largest group of side chains that are all mutually connected to each other.

In most methods, the search is over a well-defined set of rotamers for each residue. As described above, these represent local minima on the side-chain conformational potential energy map. In several methods, however, nonrotamer positions are sampled. Summers and Karplus used CHARMM to calculate potential energy maps for side chains based on 10° grids [203,204]. Dunbrack and Karplus used CHARMM to minimize the energy of rotamers

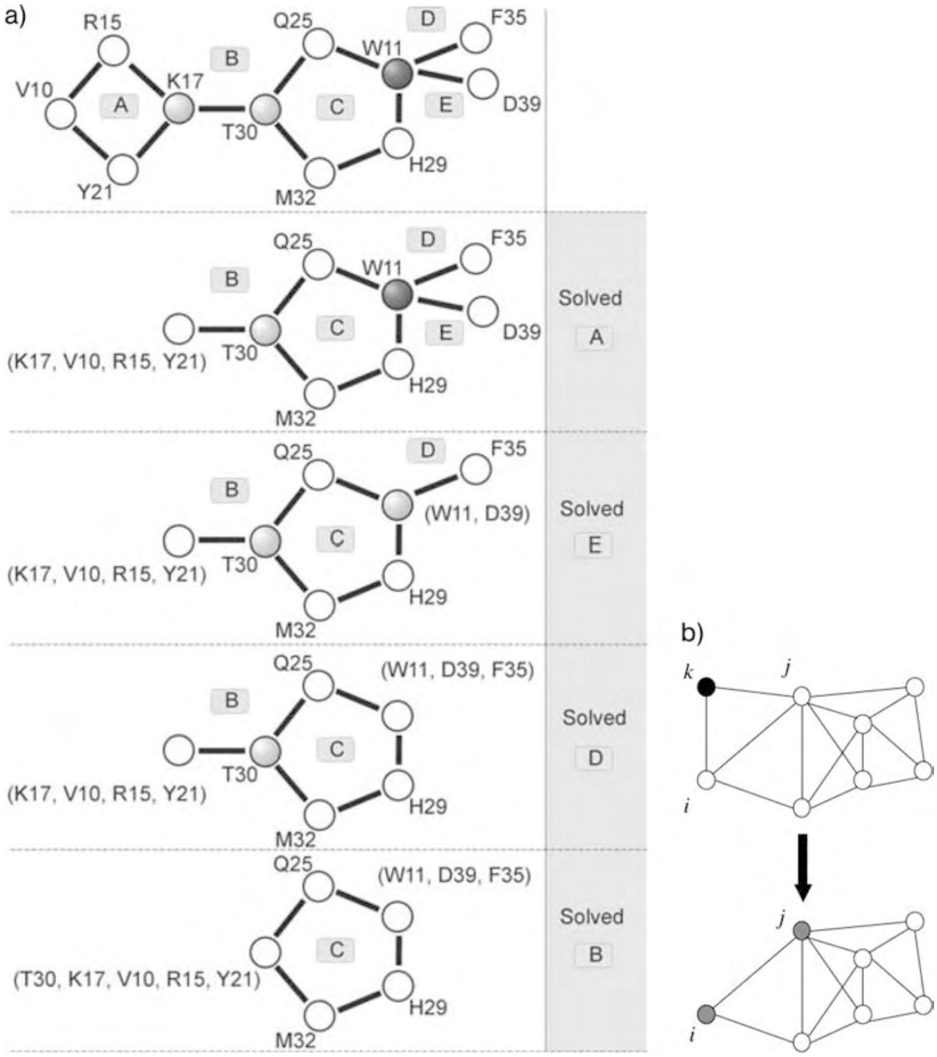


Figure 4 (a) Graph algorithm used in SCWRL3.0, solving a cluster using biconnected components. The minimum energy configuration of the cluster shown in Figure 1 is identified by stepwise solution of biconnected components. Each biconnected component is solved as shown in the right margin and the collapsed component is shown as superresidues in curly brackets. (b) Collapsing a node onto an edge.

from canonical starting conformations (–60°, 180° and +60°) [51]. Vasquez also used energy minimization [213], while Lee and Subbiah used a search over 10° increments in dihedral angles with a simple van der Waals term and a 3-fold alkane potential on side-chain dihedrals [112]. Mendes and coworkers [133, 134] used a mean-field method to sample from Gaussian distributions about the conformations in the rotamer library of Tuffery and coworkers [212].

3.2.4 Available Programs for Side-chain Prediction

While many methods for side-chain prediction have been presented over the years, only a small number of programs are publicly available at this time. Information on obtaining these programs is given in Table 2. We define "available" as either being downloadable (in source or executable form or both) from the Internet or able to be run from a webserver. Some authors will also provide their programs on request, but these programs do not generally have documentation nor are they designed for general use. They are not listed in Table 2.

3.3 Loop Modeling

3.3.1 Input Information

In stepwise construction methods, backbone segments that differ in length between the template and target (according to the sequence alignment) need to be rebuilt. In some situations, even when the sequence length of a coil segment is maintained, it may be necessary to consider alternative conformations to accommodate larger side chains or residues with differing backbone conformational requirements, Gly ↔ non-Gly or Pro ↔ non-Pro mutations. Most such loop construction methods have been tested only on native structures from which the loop to be built has been removed. However, the reality in homology modeling is more complicated, requiring several choices to be made in building the complete structure. These include how much of the template structure to remove before loop building, whether to model all side chains of the core before rebuilding the loops, and whether to rebuild multiple loops simultaneously or serially.

Deciding how much of the template structure to remove before loop building depends on examination of the sequence alignment and the template structure [114, 230]. Sequence alignments with insertions and deletions are usually not unambiguous. Most sequence alignment methods ignorant of structure will not juxtapose a gap in one sequence immediately adjacent to a gap in another sequence, i.e. they will produce an alignment that looks like this alignment:

```
AGVEPMENYKLS
SG---LDDFKLT
```

rather than like this one:

```
AGVEPMEN---YKLS
SGL-----LDDFKLT
```

However, the latter alignment is probably more realistic [1], indicating that a five-amino-acid loop in the first sequence and structure is to be replaced with a three-amino-acid loop in the second sequence. The customary practice is to remove the whole segment between two conserved secondary structures units. Even with this practice, ambiguity remains, since the ends of secondary structures, especially α-helices, are not well determined. If loop-building methods were accurate, then removing more of the segment would be a good idea. However, long loops (longer than seven amino acids) are difficult to rebuild accurately and hence there is cause to preserve as much of the starting structure as possible. Once the backbone has been borrowed from the template in stepwise modeling, one has to decide the order of building the core side chains, the backbone of loops to be built and their side chains. They may be built sequentially or allowed to vary simultaneously. Side chains from the core may guide the building of the loop, but at the same time may hinder correct placement. It is certainly the case that in the final structure there must be a reasonably low-energy conformation that can accommodate all loops and side chains simultaneously. Different authors have made different choices, and there has been little attempt to vary the procedure while keeping the search algorithm and potential energy function used fixed.

3.3.2 Loop Conformational Analysis

Loop structure prediction is always based in one way or another on an understanding of loop conformations in experimentally determined structures. Loop conformational analysis has been performed on a number of levels, ranging from classification of loops into a number of distinct types to statistical analysis of backbone dihedral angles. Loop classification schemes have usually been restricted to loops of a particular size range: short loops of one to four residues, medium loops of five to eight residues and long loops of nine residues or longer.

Thornton and coworkers have classified β-turns, which are short loops of two to five residues that connect two antiparallel β-sheet strands [195, 196, 226, 227]. These loops occur in a limited number of conformations that depend on the sequence of the loop, especially on the presence of glycine and proline residues at specific positions. The backbone conformation can be characterized by the conformations of each amino acid in terms of regions of the Ramachandran map occupied (usually defined as α_R, β_P, β_E, γ_R, α_L and γ_L) [227]. Usually one or more positions in the loops require an α_L conformation and therefore a glycine, asparagine or aspartic acid residue. One useful aspect of this analysis is that if a residue varies at certain positions or there are short insertions at certain positions, the effect on the loop can be predicted [196] since the number of possibilities for each length class is small. The programs BTPRED [192] and BHAIRPRED [103] are available (see Table 3) to predict the locations of specific types of β-turns from protein sequences and secondary structure predictions. Single-amino-acid changes tend to maintain the loop conformations, except when Pro residues substitute for residues with $\phi > 0^\circ$, while insertions change the class of the loop.

Table 3 Publicly available loop conformation prediction programs

Program	Availability	Website
Rapper	web, download	http://raven.bioc.cam.ac.uk
ModLoop	web	http://alto.compbio.ucsf.edu/modloop//modloop
Loopy	download	http://honiglab.cpmc.columbia.edu/programs/loop
PLOP	download	http://francisco.compbio.ucsf.edu/~jacobson/plop_manual/plop_overview
MODELLER	download	http://salilab.org/modeller

In recent years with a larger number of structures available, medium-length loops have also been classified [38,44,56,69,104,115,135,147,149,150,229] by their patterns of backbone conformation residue by residue (α_R, β_P, etc.). A number of regularly occurring classes have been found, depending on length, type of secondary structure being connected and sequence. These classes cover many but by no means all of the loops seen in non-β turn contexts.

Longer loops (with more than eight amino acids) have been investigated by Martin and coworkers [131] and Ring and coworkers [165]. Martin and coworkers found that long loops fall into two classes: those that connect spatially adjacent secondary structures and those that connect secondary structures separated by some distance. Ring and coworkers provided a useful classification of longer loops as either strap (long extended loops), Ω loops (similar to those described by Leszczynski [115] and Pal and coworkers [150]), which resemble the Greek letter, and ζ loops, which are nonplanar and have a zigzag appearance. The different loop types were found to have different distributions of virtual C_α–C_α–C_α–C_α dihedrals to accommodate their shapes.

A number of groups have updated the Ramachandran propensities of the 20 amino acids. Swindells and coworkers [209] have calculated the intrinsic ϕ,ψ propensities of the 20 amino acids from the coil regions of 85 protein structures. The distribution for coil regions is quite different than for the regular secondary structure regions, with a large increase in β_P and α_L conformations, and much more diverse conformations in the β_E and α_R regions. Their results also indicate that the 18 non-Gly,Pro amino acid type are in fact quite different from each other in terms of their Ramachandran distributions, despite the fact that they are often treated as identically distributed in prediction methods [25, 57]. Their analysis was divided into the main broad regions of the Ramachandran map, ignoring the α_L region. The results are intriguing, in that the probability distributions are distinct enough even when calculated from a relatively small protein dataset. More recently Lovell and coworkers [124] and Anderson and coworkers [4] have produced new Ramachandran maps based on stricter criteria for inclusion of amino acids based on resolution, R-factors and B-factors, as well as data smoothing techniques that remove outliers and unpopulated parts of the Ramachandran map. Their results indicate that a stricter adherence to “allowed” regions is called for, since nearly all residues in disallowed regions are based on poor electron density.

3.3.3 **Loop Prediction Methods**

Loop prediction methods can be analyzed for a number of important factors in determining their usefulness: (i) method of backbone construction, (ii) what range of lengths are possible, (iii) how widely is the conformational space searched, (iv) how are side chains added, (v) how are the conformations scored (i.e., the potential energy function) and (vi) how much has the method been tested (length, number, self/nonself).

The most common approach to loop modeling involves using “spare parts” from other (unrelated) protein structures [10, 32, 60, 61, 63, 72, 81, 91, 96, 114, 135, 165, 168, 172, 205, 207, 217, 230, 231]. These database methods begin by measuring the orientation and separation of the backbone segments flanking the region to be modeled, and then search the PDB for segments of the same length that span a region of similar size and orientation. This work was pioneered by Jones and Thirup [91]. They defined a procedure in which C_α–C_α distances were measured among six residues, three on either side of a backbone segment to be constructed. These 15 C_α–C_α distances were used to search structures in the PDB for segments with similar C_α–C_α distances and the appropriate number of intervening residues. Other authors have used the same method for locating potential database candidates for the loop to be constructed [60, 96, 205, 217]. The fragment selection method used in Rosetta *ab initio* modeling [197] is based at least in part on the database approach to loop modeling, and is used in Rosetta for loop construction in homology modeling [167]. In recent years, as the size of the PDB has increased, database methods have continued to attract attention. With a larger database, recurring structural motifs have been classified for loop structures [44, 56, 104, 113, 135, 147, 172], including their sequence dependence.

Although many methods have been published, they have usually only been tested on a small number of loops, and then usually in the context of rebuilding loops onto their own backbones, rather than in the process of homology modeling. A recent exception is that of Fernandez-Fuentes and coworkers [61] who tested the ArchDB database [56] of loops as a predictive tool. They used a “jackknife” test that removed all loops from the same superfamily for each loop in a set of over 10 000 used to construct ArchDB.

The main alternative to database methods is *ab initio* construction of loops by random or exhaustive search mechanisms. These methods are quite varied in their generation and subsequent modification of loop structures to fit the environment of the fitted segment. The initial conformation may be random, starting from the N- or C-terminal anchor, so that the other end of the loop does not connect to the other anchor (the C- or N-terminal anchor, respectively). Such loops can then be closed using energy minimization that places some energetic constraints on a closed loop, or using loop closure methods, such as those based on inverse kinematics in robotics [19, 28, 35]. Other methods have built chains by sampling Ramachandran conformations randomly, keeping partial segments as long as they can complete the loop with the remaining residues to be built [64, 191, 198].

An alternative approach to the loop generation problem is to use a geometrically distorted loop that bridges the two anchors exactly and then to relax the structure into an undistorted protein-like structure. MODELLER starts loop modeling with a linear arrangement of the atoms in the loop, which is then relaxed into a protein-like conformation using energy minimization [65]. Zheng and coworkers used a scaling-relaxation method in which an initially generated or database loop is scaled in size until it fits the anchors [244–246]. This results in very short bond distances and unphysical connections to the anchors. From there, energy minimization is performed on the loop, slowly relaxing the scaling constant, until the loop is scaled back to full size.

One important aspect in the development of a prediction method based on random (or exhaustive) construction of backbone conformations is the free energy function used to discriminate among those conformations that successfully bridge the anchors. Fogolari and Tosatto have found that a free energy function including a molecular mechanics potential energy and a Poisson–Boltzmann solvent-accessible surface area solvation term was able to identify decoys from a large set that were close to the native structure [66]. Jacobson and coworkers recently used the OPLS (optimized potential for liquid simulations) molecular mechanics force field, with improved torsional energy parameters optimized to reproduce quantum-mechanical data and side-chain prediction [87], in combination with a surface-generalized Born/nonpolar (SGB/NP) hydration free energy model [68]. Their search method generated one residue at a time from a 5° resolution backbone model with steric and side-chain checks, from both ends of the loop, followed by clustering and energy minimizations of cluster representatives. Their method was tested on a large set of 833 loops with excellent results for loops up to 12 residues in length.

3.3.4 **Available Programs**

Very few loop modeling programs *per se* are publicly available, although loop modeling is integral to more complete modeling programs. A list of available loop modeling programs is given in Table 3. Some programs that do complete modeling but can be used for loop modeling without further refinement are listed (e.g. MODELLER).

3.4 Methods for Complete Modeling

Homology modeling is a complex process. Automated protocols that begin with a sequence and produce a complete model are few, and the resulting models should be examined with great care (as of course should all models). However, these methods usually allow for (and indeed recommend) some manual intervention in the choice of template structure or structures and in the sequence alignment. In these steps, manual intervention is likely to have important consequences. Later stages of modeling (actual building of the structure) are more easily automated and there are not usually obvious manual adjustments to make.

There are several publicly available programs available for homology modeling that are intended to make complete models from input sequences. These include MODELLER [174, 175, 180, 181], RAMP [176–178] and MolIDE [29]. There are also several webservers that provide homology modeling services, including SWISS-MODEL [74, 153, 154], Esypred [105] and 3D-JIGSAW [11]. Program availability is given in Table 4. Some of these programs provide only BLAST/PSI-BLAST searching followed by model-building with MODELLER (e.g. EsyPRED). We describe some of these programs.

Table 4 Publicly available comparative modeling programs

Program	Availability	Website
3d-JIGSAW	web	http://www.bmm.icnet.uk/servers/3djigsaw
CPHmodels	web	http://www.cbs.dtu.dk/services/CPHmodels
EsyPred	web	http://www.fundp.ac.be/urbm/bioinfo/esypred
FAMS	E-mail server	http://www.pharm.kitasato-u.ac.jp/fams
Geno3D	web	http://geno3d-pbil.ibcp.fr
MODELLER	download	http://salilab.org/modeller
ModWeb	web	http://salilab.org/modweb
Modzinger	web	http://peyo.ulb.ac.be/mz/index
nest	download	http://honiglab.cpmc.columbia.edu/programs/nest
parmodel	web	http://laboheme.df.ibilce.unesp.br/cluster/parmodel_mpi
Robetta	web	http://robetta.bakerlab.org
SDSC	web	http://cl.sdsc.edu/hm
SWISS-MODEL	web	http://swissmodel.expasy.org//SWISS-MODEL

3.4.1 MODELLER

MODELLER takes as input a protein sequence and a sequence alignment to the sequence(s) of known structure(s), and produces a comparative model. The program uses the input structure(s) to construct constraints on atomic distances, dihedral angles, etc., that when combined with statistical distributions derived from many homologous structure pairs in the PDB form a conditional probability distribution function for the degrees of freedom of the protein.

For instance, a probability function for the backbone dihedrals of a particular residue to be built in the model can be derived by combining information in the known structure (given the alignment) and information about the amino acid type's Ramachandran distribution in the PDB. The number of constraints is very large; for a protein of 100 residues there may be as many as 20 000 constraints. The constraints are combined with the CHARMM force field to form a function to be optimized. This function is optimized using conjugate gradient minimization and molecular dynamics with simulated annealing.

3.4.2 **MolIDE: A Graphical User Interface for Modeling**

MolIDE (Molecular Interactive Design Environment) is an open-source, extensible graphical user interface for homology modeling [29]. MolIDE provides a graphical interface for running sequence database searches with PSI-BLAST, searches of the PDB, secondary structure prediction, manual alignment editing, and running loop and side-chain prediction programs. One of MolIDE's main benefits is allowing a user to edit a sequence–structure alignment and to view the positions of insertions and deletions within the template structure in real time. MolIDE also allows manual choice of anchor residues for loop modeling with the assistance of a graphical view of the template protein structure. MolIDE runs on the Windows and Linux operating systems. The use of MolIDE will be illustrated in the next section with an example of comparative modeling of a protein of biological interest.

3.4.3 **RAMP and PROTINFO**

Samudrala and Moult described a method for "handling context sensitivity" of protein structure prediction, i.e. simultaneous loop and side-chain modeling, using a graph theory method [178, 179] and an all-atom distance-dependent statistical potential energy function [176]. These methods are also implemented in the PROTINFO webserver listed in Table 4.

3.4.4 **SWISS-MODEL**

SWISS-MODEL is intended to be a complete modeling procedure accessible via a web server that accepts the sequence to be modeled and then delivers the model by electronic mail [74, 154]. In contrast to MODELLER, SWISS-MODEL follows the standard protocol of homolog identification, sequence alignment, determining the core backbone, and modeling loops and side chains. SWISS-MODEL will search a sequence database of proteins in the PDB with BLAST, and will attempt to build a model for any PDB hits with p-values less than 10^{-5} and at least 30% sequence identity to the target. SWISS-MODEL allows for user intervention by specifying the template(s) and alignments to be used.

If more than one structure is found, the structures will be superimposed on the template structure closest in sequence identity to the target.

SWISS-MODEL determines the core backbone from the alignment of the target sequence to the template sequence(s) by averaging the structures according to their local degree of sequence identity with the target sequence. The program builds new segments of backbone for loop regions by a database scan of the PDB using anchors of four C_α atoms on each end. This method is used to build only the C_α atoms and the backbone is completed with a search of pentapeptide segments in the PDB that fit the C_α trace of the loop. Side chains are now built for those residues without information in the template structure by using the most common (backbone-independent) rotamer for that residue type. If a side chain can not be placed without steric overlaps, another rotamer is used. Some additional refinement is performed with energy minimization with the GROMOS [75] program.

4 Results

4.1 Range of Targets

A very large number of homology models have been built over the years by many authors. Recent targets have included proteins of significant interest in biology and medicine 10- [2, 9, 16, 20, 26, 27, 33, 36, 58, 67, 77, 128, 129, 145, 162, 171, 206, 216, 242]. Several databases of homology models are available on the Internet, including ModBase [158], FAMSBASE [237] and the SWISS-MODEL repository [100]. Their websites are given in Table 4.

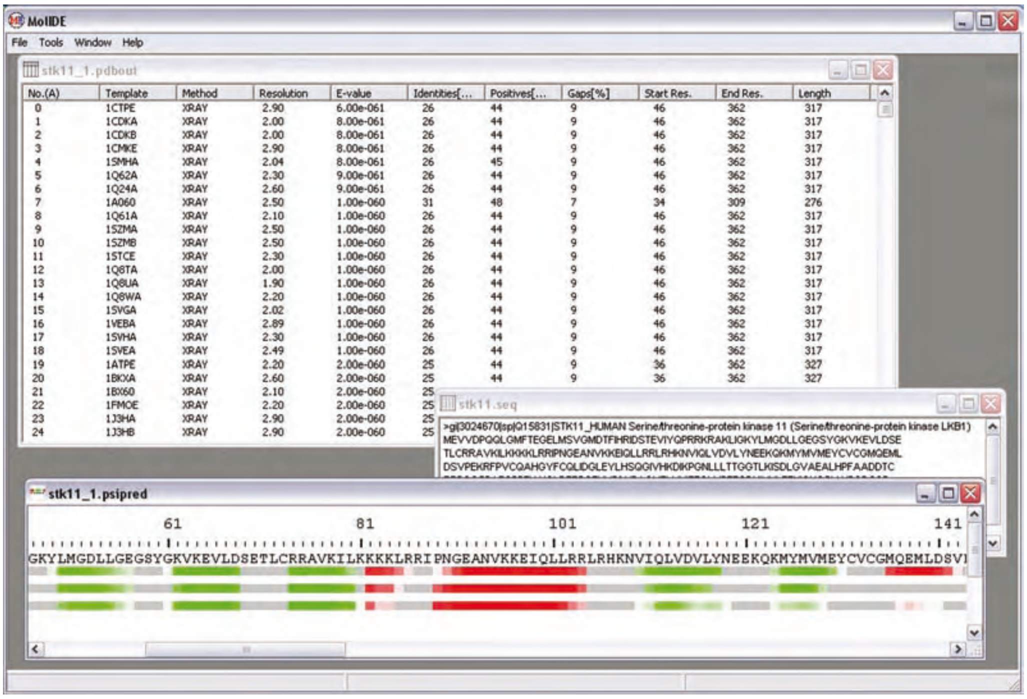
4.2 Example: Protein Kinase STK11/LKB1

The protein kinase STK11 is frequently mutated in human cancers and mutations in this gene are strongly associated with Peutz–Jeghers syndrome [79, 89]. Patients with Peutz–Jeghers syndrome often develop dark spots on the lips and inside the mouth as well as near the eyes and nostrils. These patients also develop polyps in the stomach and intestine, and are very susceptible to cancers of the breast, colon, pancreas, stomach and ovary [188]. This disease is inherited in an autosomal dominant manner, so that a mutation in a single copy of the gene is enough to confer risk [110].

One important use of homology modeling is to understand how missense mutations may lead to disease. In general, missense mutations that have deleterious effects lead to amino changes that either affect stability or dynamics of a folded protein, or affect interactions of the protein product with other molecules, including other proteins, DNA or ligands. While most of the mutations associated with the disease lead to a truncated protein, many missense mutations have also been linked to Peutz–Jeghers syndrome [110]. It is therefore of interest to build a model of the STK11 kinase domain and examine the location and likely effects of disease-associated mutations.

As described above, MolIDE [29] assists a user in modeling a protein by providing a graphical interface to the steps involved in basic homology modeling, including sequence database searching, alignment editing, and loop and side-chain modeling. We obtained the sequence of STK11 from the NCBI website in FASTA format, as shown in the middle panel of Figure 5. Once this sequence is input into MolIDE, the user runs PSI-BLAST from the Tools menu. PSI-BLAST is set up to run several rounds of search against the nonredundant protein database from NCBI. The version of PSI-BLAST distributed with MolIDE has been modified to output a profile matrix with a unique name after each iteration of the search so that each matrix can then be used to search the PDB sequence database included with MolIDE (G. Wang and R.L. Dunbrack, Jr., unpublished). These profiles are also used by the secondary structure prediction program PSIPRED [90], included with MolIDE. The secondary structure predictions are shown in the lower panel of Figure 5, where the red and green colors indicate α-helices and β-sheet strands, respectively, and the intensity of the color represents the confidence level of the prediction. As the profile includes more and more sequences remotely related to the query, the secondary structure prediction also changes. For some proteins, the prediction gets better as the signal from the multiple alignment becomes stronger, while for others the prediction may worsen if many sequences with variations in secondary structure (longer loops, shorter or longer secondary structure elements) get aligned or even misaligned to the profile.

As STK11 is a kinase, we have a large variety of structures in the PDB that can be used as a template. It is important to make a good choice, since the quality of the model will depend on the template or templates used. Currently, MolIDE does not model from multiple templates, so we need to select one structure. However, we could make a number of models based on different templates and compare them. A list of hits from the PDB for STK11 is shown in the upper panel of Figure 5. This list was generated from a PSI-BLAST search using the profile generated from the first round of PSI-BLAST on the nonredundant database. Information about each template, including PDB code, experiment type, resolution, *E*-value, sequence identity, length of template, starting positions of alignment and length of alignment, is given. Template choice is facilitated in this table by sorting based on any of the categories by clicking on the column header. Three different sorts of the table are shown in Figure 6, sorted by end query residue of the alignment (top), resolution (middle) and percent gaps in the alignment (bottom).



No.(A)	Template	Method	Resolution	E-value	Identities[...	Positives[...	Gaps[%]	Start Res.	End Res.	Length
0	1CTPE	XRAY	2.90	6.00e-061	26	44	9	46	362	317
1	1CDKA	XRAY	2.00	8.00e-061	26	44	9	46	362	317
2	1CDKB	XRAY	2.00	8.00e-061	26	44	9	46	362	317
3	1CMKE	XRAY	2.90	8.00e-061	26	44	9	46	362	317
4	1SMHA	XRAY	2.04	8.00e-061	26	45	9	46	362	317
5	1Q62A	XRAY	2.30	9.00e-061	26	44	9	46	362	317
6	1Q24A	XRAY	2.60	9.00e-061	26	44	9	46	362	317
7	1A060	XRAY	2.50	1.00e-060	31	48	7	34	309	276
8	1Q61A	XRAY	2.10	1.00e-060	26	44	9	46	362	317
9	1SZMA	XRAY	2.50	1.00e-060	26	44	9	46	362	317
10	1SZMB	XRAY	2.50	1.00e-060	26	44	9	46	362	317
11	1STCE	XRAY	2.30	1.00e-060	26	44	9	46	362	317
12	1Q8TA	XRAY	2.00	1.00e-060	26	44	9	46	362	317
13	1Q8UA	XRAY	1.90	1.00e-060	26	44	9	46	362	317
14	1Q8WA	XRAY	2.20	1.00e-060	26	44	9	46	362	317
15	1SVGA	XRAY	2.02	1.00e-060	26	44	9	46	362	317
16	1VEBA	XRAY	2.89	1.00e-060	26	44	9	46	362	317
17	1SVHA	XRAY	2.30	1.00e-060	26	44	9	46	362	317
18	1SVEA	XRAY	2.49	1.00e-060	26	44	9	46	362	317
19	1ATPE	XRAY	2.20	2.00e-060	25	44	9	36	362	327
20	1BKXA	XRAY	2.60	2.00e-060	25	44	9	36	362	327
21	1BX60	XRAY	2.10	2.00e-060	25					
22	1FMOE	XRAY	2.20	2.00e-060	25					
23	1J3HA	XRAY	2.90	2.00e-060	25					
24	1J3HB	XRAY	2.90	2.00e-060	25					

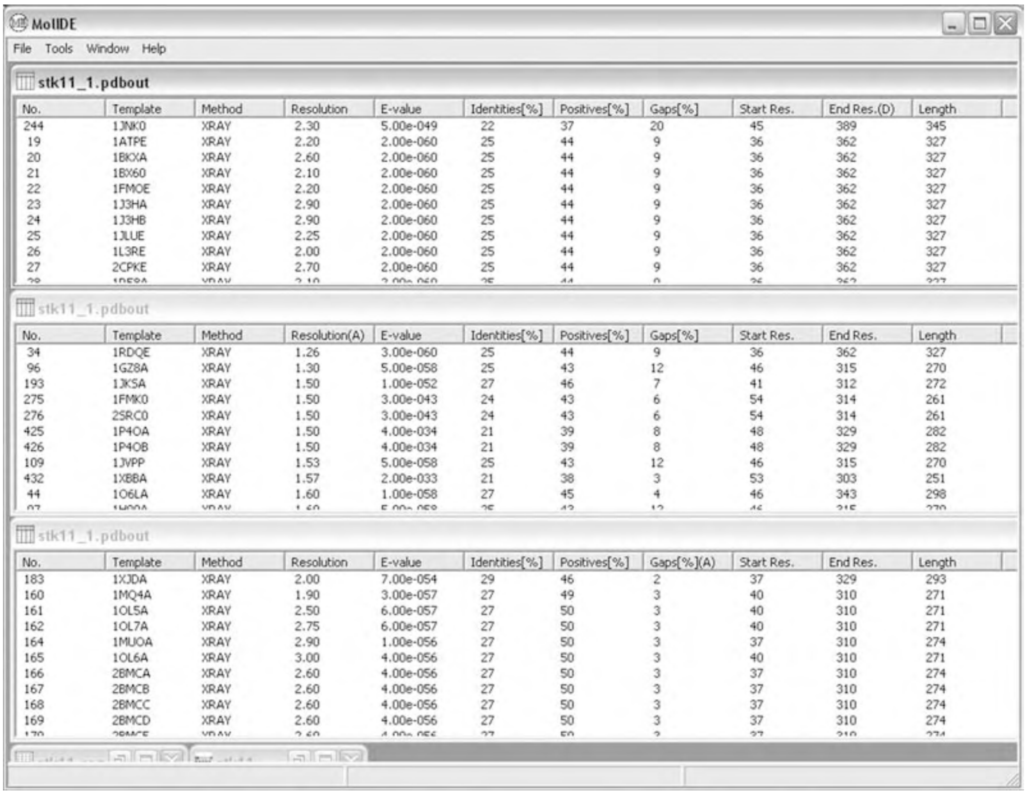
```
>gi|3024670|sp|Q15831|STK11_HUMAN Serine/threonine-protein kinase 11 (Serine/threonine-protein kinase LKB1)
MEVVDPQQLGMFTEGELMSVGMDTFIHRIDSTEVIYQPRRKRAKLIGKYLMGDLLGEGSYGKVKEVLDSE
TLCRRAVKILKKKKLRRIPNGEANVKKEIQLLRRLRHKNVIQLVDVLYNEEKQKMYMVMEYCVCGMQEML
DSVPEKRFPVCQAHGYFCQLIDGLEYLHSQGIVHKDIKPGNLLLTTGGTLKISDLGVAEALHPFAADDTC
```

```
              61                  81                  101                 121                 141
GKYLMGDLLGEGSYGKVKEVLDSETLCRRAVKILKKKKLRRIPNGEANVKKEIQLLRRLRHKNVIQLVDVLYNEEKQKMYMVMEYCVCGMQEMLDSVI
```

Figure 5 Screenshot of MolIDE for modeling STK11. Top: hits from a search of the PDB for protein target sequence STK11. Middle: sequence of STK11. Bottom: secondary structure prediction for STK11 using PSIPRED after each of three rounds of PSI-BLAST. Helix predictions are in red and sheet strand predictions are in green, and the intensity of the color is proportional to the confidence levels produced by PSIPRED.

The usefulness of a template depends on many factors. It is not necessarily the case that one should use the highest-sequence-identity hit or the best *E*-value. The number and location of gaps in the alignment should also be considered, as should the presence of desirable ligands and structure quality. For STK11, the majority of template alignments end around residue 310, which is the end of the kinase domain, but obviously it would be useful to model the region C-terminal to the kinase domain (residues 310–433).

By double-clicking on the PDB code, MolIDE opens up a window with the alignment as shown for that PDB entry. This window contains the alignment at the bottom and a rotatable view of the backbone of the template in the top part of the window (Figure 7). Conserved amino acids are indicated between the query (top) and hit (bottom) sequences. Gaps in the alignment are indicated by blue squares as are residues in the template that are missing in the coordinates due to poor electron density. Positions in the template structure where insertions need to be modeled, because the query sequence is longer in



stk11_1.pdbout

No.	Template	Method	Resolution	E-value	Identities[%]	Positives[%]	Gaps[%]	Start Res.	End Res.(D)	Length
244	1JNK0	XRAY	2.30	5.00e-049	22	37	20	45	389	345
19	1ATPE	XRAY	2.20	2.00e-060	25	44	9	36	362	327
20	1BKXA	XRAY	2.60	2.00e-060	25	44	9	36	362	327
21	1BX60	XRAY	2.10	2.00e-060	25	44	9	36	362	327
22	1FMOE	XRAY	2.20	2.00e-060	25	44	9	36	362	327
23	1J3HA	XRAY	2.90	2.00e-060	25	44	9	36	362	327
24	1J3HB	XRAY	2.90	2.00e-060	25	44	9	36	362	327
25	1JLUE	XRAY	2.25	2.00e-060	25	44	9	36	362	327
26	1L3RE	XRAY	2.00	2.00e-060	25	44	9	36	362	327
27	2CPKE	XRAY	2.70	2.00e-060	25	44	9	36	362	327

stk11_1.pdbout

No.	Template	Method	Resolution(A)	E-value	Identities[%]	Positives[%]	Gaps[%]	Start Res.	End Res.	Length
34	1RDQE	XRAY	1.26	3.00e-060	25	44	9	36	362	327
96	1GZ8A	XRAY	1.30	5.00e-058	25	43	12	46	315	270
193	1JKSA	XRAY	1.50	1.00e-052	27	46	7	41	312	272
275	1FMK0	XRAY	1.50	3.00e-043	24	43	6	54	314	261
276	2SRC0	XRAY	1.50	3.00e-043	24	43	6	54	314	261
425	1P4OA	XRAY	1.50	4.00e-034	21	39	8	48	329	282
426	1P4OB	XRAY	1.50	4.00e-034	21	39	8	48	329	282
109	1JVPP	XRAY	1.53	5.00e-058	25	43	12	46	315	270
432	1XBBA	XRAY	1.57	2.00e-033	21	38	3	53	303	251
44	1O6LA	XRAY	1.60	1.00e-058	27	45	4	46	343	298

stk11_1.pdbout

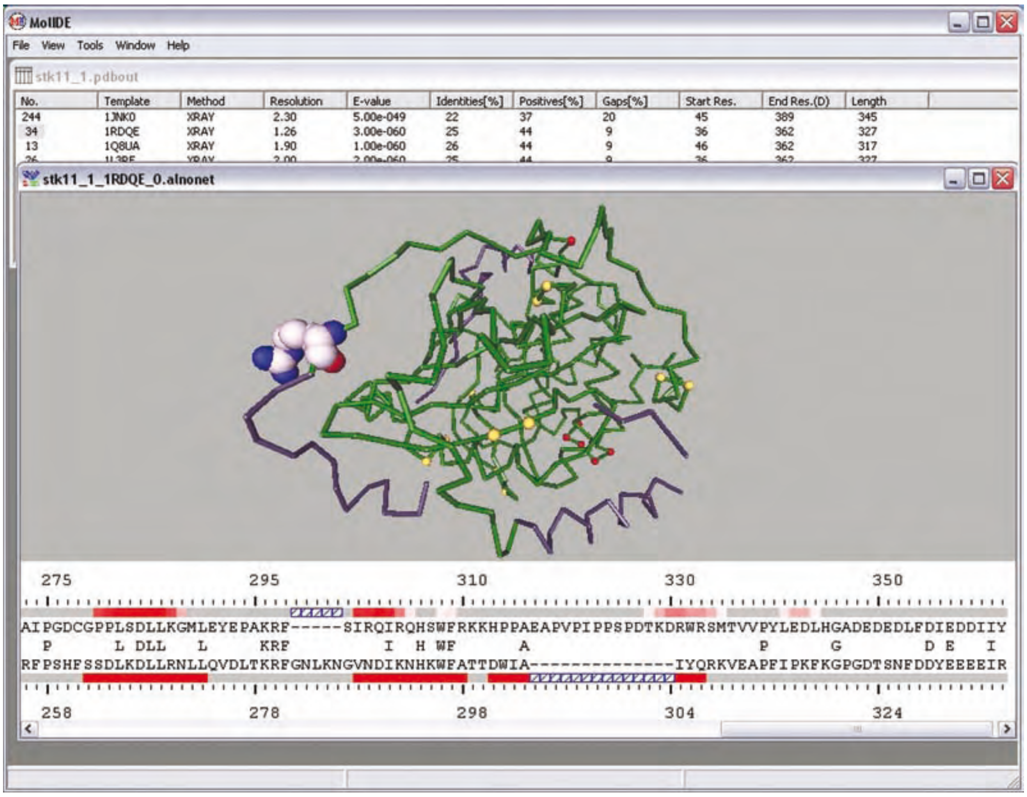
No.	Template	Method	Resolution	E-value	Identities[%]	Positives[%]	Gaps[%](A)	Start Res.	End Res.	Length
183	1XJDA	XRAY	2.00	7.00e-054	29	46	2	37	329	293
160	1MQ4A	XRAY	1.90	3.00e-057	27	49	3	40	310	271
161	1OL5A	XRAY	2.50	6.00e-057	27	50	3	40	310	271
162	1OL7A	XRAY	2.75	6.00e-057	27	50	3	40	310	271
164	1MUOA	XRAY	2.90	1.00e-056	27	50	3	37	310	274
165	1OL6A	XRAY	3.00	4.00e-056	27	50	3	40	310	271
166	2BMCA	XRAY	2.60	4.00e-056	27	50	3	37	310	274
167	2BMCB	XRAY	2.60	4.00e-056	27	50	3	37	310	274
168	2BMCC	XRAY	2.60	4.00e-056	27	50	3	37	310	274
169	2BMCD	XRAY	2.60	4.00e-056	27	50	3	37	310	274

Figure 6 Choosing templates for modeling STK11. Screenshot of MolIDE showing PDB hits sorted by end residue (top), resolution (middle) and percent gaps in alignment (bottom).

that region, are marked with yellow spheres on the structure view. Deletions from the structure are shown on the protein structure as red spheres.

Figure 7 shows the C-terminal end of the alignment for template 1RDQ (chain E) [238]. It is fairly clear from this view that the extension of the alignment beyond residue 316 of the query shows very little similarity in sequence with the template and includes a very large insertion that would need to be modeled. It is likely that this region is incorrectly aligned. This often happens with PSI-BLAST type alignments, such that alignments extend beyond conserved domains due to chance similarities, especially in regions without significant regular secondary structure.

After examining a number of the other templates that extend well beyond residue 310, it was clear that none of them gave a very good alignment for the C-terminal portion of STK11. While a large number of the templates share similar sequence identity to STK11 on the order of 21–27%, some of them contain substantially fewer gaps in the alignment than others. Sorting by



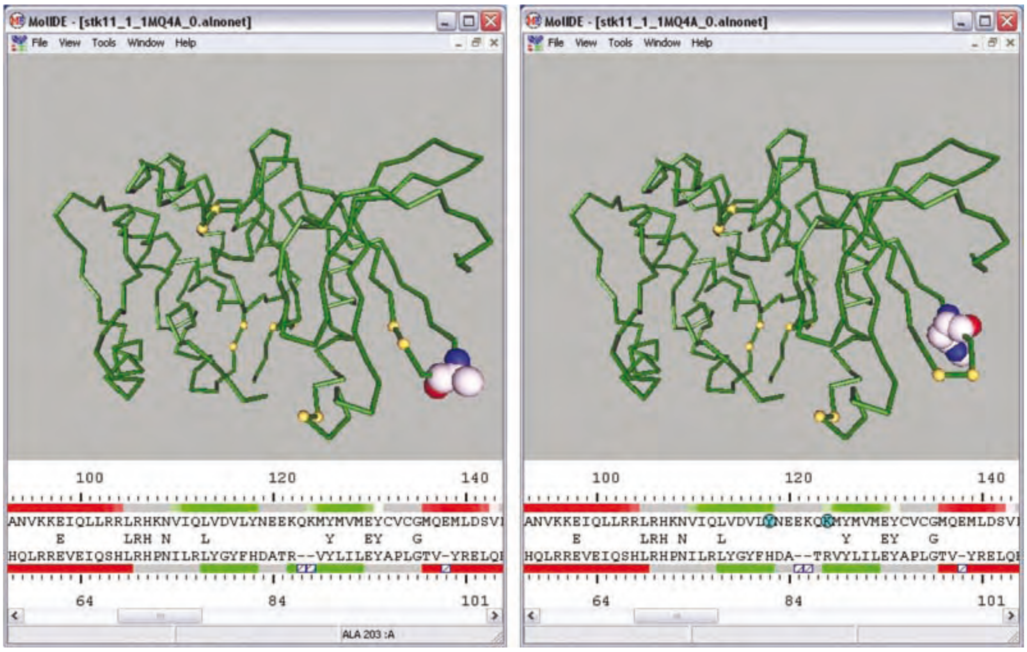
No.	Template	Method	Resolution	E-value	Identities[%]	Positives[%]	Gaps[%]	Start Res.	End Res.(D)	Length
244	1JNK0	XRAY	2.30	5.00e-049	22	37	20	45	389	345
34	1RDQE	XRAY	1.26	3.00e-060	25	44	9	36	362	327
13	1Q8UA	XRAY	1.90	1.00e-060	26	44	9	46	362	317
26	1L3RE	XRAY	2.00	2.00e-060	25	44	9	36	362	327

```
 275                 295            310                 330                 350
AIPGDCGPPLSDLLKGMLEYEPAKRF-----SIRQIRQHSWFRKKHPPAEAPVPIPPSPDTKDRWRSMTVVPYLEDLHGADEDEDLFDIEDDIIY
  P      L DLL   L     KRF         I  H WF      A                      P      G        D E   I
RFPSHFSSDLKDLLRNLLQVDLTKRFGNLKNGVNDIKNHKWFATTDWIA--------------IYQRKVEAPFIPKFKGPGDTSNFDDYEEEEIR
 258                 278            298                 304                 324
```

Figure 7 Template 1RDQ (chain E) as template for STK11. The end portion of the alignment is shown, indicating that the residues after STK11 residue 310 are poorly aligned with little sequence similarity and a large gap. Gaps in the alignment are marked with blue squares, and conserved residues are marked between the query (top) and hit (bottom) sequences. The predicted secondary structure is above the query (red = helix; green = sheet) and the experimental secondary structure of the hit is below the template sequence. In the structure view, residues deleted from the structure are indicated with red balls and points of insertion are marked by pairs of yellow balls. The aligned portion of the template is in green and the unaligned portions are in gray. The last residue of the aligned portion is in spacefill representation (Tyr336).

percent gaps, we find a group of templates that have 2–3% of the alignment as gaps for residues 40–310, as shown in the bottom panel of Figure 6. We chose as template PDB entry 1MQ4, a 1.9-Å structure of aurora-A kinase [143], since this structure was the highest resolution of these and contained ADP in the active site of the kinase.

One of the benefits of MolIDE in user-assisted homology modeling is the ability to edit the sequence with the assistance of a graphical view of the protein with the locations of insertions and deletions. For instance, in Figure 8 the alignments before (left screenshot) and after (right screenshot) editing are shown. In the left figure, the two-residue insertion (residues 123–124 with


MolIDE - [stk11_1_1MQ4A_0.alnonet]
File View Tools Window Help
ALA 203 :A
MolIDE - [stk11_1_1MQ4A_0.alnonet]
File View Tools Window Help

```
        100                 120                 140
ANVKKEIQLLRRLRHKNVIQLVDVLYNEEKQKMYMVMEYCVCGMQEMLDSVI
     E      LRH N   L            Y   EY   G
HQLRREVEIQSHLRHPNILRLYGYFHDATR--VYLILEYAPLGTV-YRELQI
        64                  84                101
```

```
        100                 120                 140
ANVKKEIQLLRRLRHKNVIQLVDVLYNEEKQKMYMVMEYCVCGMQEMLDSVI
     E      LRH N   L            Y   EY   G
HQLRREVEIQSHLRHPNILRLYGYFHDA--TRVYLILEYAPLGTV-YRELQI
        64                  84                101
```

Figure 8 Manual editing of sequence alignments based on the structure view of template 1MQ4 for STK11. The alignment before editing is shown at left and after editing at right. The spacefilled residues mark the loop being edited (Ala84 left and His82 right). Note that the positions of the yellow balls marking the position of the insertion move as the alignment is edited. The anchor positions for loop modeling set manually are indicated on the right (Tyr118 and Lys124 of STK11).

sequence QK) occurs inside a β-sheet strand. Ala84 in the middle of the neighboring loop is shown in spacefill representation to mark the location on the structure. After editing the alignment by "ctrl"-clicking and "shift"-clicking to delete the gap and to create it in another location, the alignment appears as it does in the right side of the figure (with the last residue of the sheet strand, His82, in spacefill this time).

Once the sequence alignment is edited, e.g. by moving insertions and deletions into the middle of loop regions as described above, loop and side-chain modeling commands can be called from the menus. Our side-chain modeling program SCWRL is integrated into MolIDE and Loopy is used for loop modeling [235]. To model loops, the positions of left and right anchor residues are set by pointing and "right"-clicking on the query sequence and then calling Loopy from the Tools menu. The anchors for the loop consisting of residues 118–123 are shown in the right screenshot in Figure 8.

After modeling each of the five insertions and adding the coordinates of the ADP and magnesium ions, we can view the structure superposed on its template as shown in Figure 9. MolIDE is not set up to model in the presence

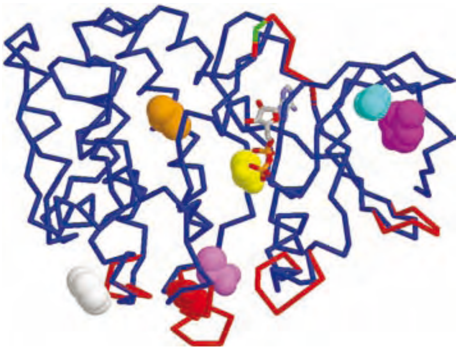


Figure 9 Model of STK11. The template backbone is in blue and the modeled loops for STK11 are in red. The ADP bound in the active site is in stick figure and CPK coloring (carbon = gray; oxygen = red; nitrogen = blue; phosphorus = orange). Several mutations associated with development of cancer are shown in spacefill representation, including Tyr49 (magenta), Val66 (cyan), Leu160 (orange), Asp194 (yellow), Glu199 (violet), Asp208 (red) and Phe231 (white). Gly135 is marked on the backbone in green.

of the ligand, although this can be accomplished for side chains with SCWRL outside of MolIDE. Most of the insertions are some distance away from the active site and four of these are relatively close in space to each other, all on the bottom face of the protein as oriented in Figure 9.

The positions of some missense mutations associated with cancer are marked with spacefill on the structure and three of these mutations are located in the modeled loops at the bottom of the protein. Another one is in the modeled loop at the top. Three mutations are buried in the hydrophobic core, of which two are in the N-terminal domain and one is in the C-terminal domain (colored in cyan, magenta and orange, respectively). These mutations are likely to disturb the hydrophobic cores of these regions, leading to instability of the folded structure. One additional mutation, in yellow spheres, is an Asp residue that binds the magnesium ions which stabilize the binding of ATP to the kinase. Loss of this Asp is likely to result in loss of magnesium and inability to bind ATP. STK11 binds a number of other proteins, and it is possible that one of these proteins binds to the modified loops at the bottom of the structure and that mutations in these loops leads to lack of binding of important interactors of STK11. Mutations which affect binding of the STRAD protein have been analyzed using a homology model of STK11 by Boudeau and coworkers [21]. While STRAD is homologous to kinase domains, none of the available dimeric kinase templates appears to have a binding interface consistent with these mutations, indicating that perhaps STRAD does not bind to STK11 in a manner similar to existing dimer interfaces of kinases.

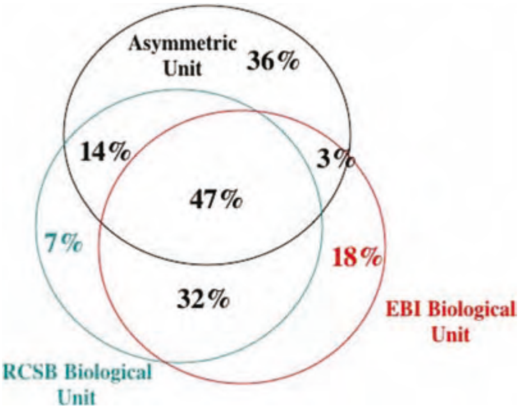


Figure 10 Venn diagram of similarities among asymmetric units, RCSB biological units and EBI (PQS) biological units. Each circle represents 24 000 entries available from both RCSN and PQS sites. Areas are only approximations to percentages marked in each overlapping or nonoverlapping region. For instance, 36% of the 24 000 entries have an asymmetric unit that is different from both RCSB and PQS.

4.3 The Importance of Protein Interactions

The STK11 example points out the importance of incorporating information on protein interactions in homology modeling. As described above, the Research Collaboratory for Structural Bioinformatics (RCSB) provides structures of the asymmetric unit, rather than the biological unit for crystal structures. RCSB also provides separate files that contain the proposed biological unit(s) for each structure, which may be larger or smaller than the asymmetric unit. To quantify this issue, we compared the asymmetric units and the biological units as provided by both the RCSB and the EBI/macromolecular structure database (MSD) [80]. For 23 418 structures available in PQS, Figure 10 shows the similarity of these three sets of units. Figure 10 shows that 53% of asymmetric units are different from either the RCSB or the PQS biological unit or both. This indicates that the standard entry from the PDB is not the biological unit at least half of the time and that the other two sources should be consulted. In addition, RCSB and PQS do not agree on biological units 21% of the time. Unfortunately, there are no automated ways to model homo-multimers, other than to model the sequence on each chain of a known multimer structure. SWISS-MODEL does provide a way to combine models made from different chains of a template biological unit file from PQS.

To illustrate the importance of these interactions, we investigated the large set of mutations in Lac Repressor investigated by Miller and colleagues [130, 148, 202], These authors presented functional data on 4042 mutations of the *Escherichia coil* Lac repressor. The Lac repressor function was evaluated *in vivo* by observing expression levels of β-galactosidase with and without allosteric induction by isopropyl-β-D-galactoside (IPTG). Thus mutations that reduce stability of the Lac repressor monomer, the affinity of monomers in the tetramer, as well as those that affect binding to DNA will produce a visible blue phenotype by the expression of β-galactosidase. Mutations that affect binding of IPTG will not be inducible and thus remain as white colonies, even in the presence of IPTG. This is an ideal system for identifying parameters that can be used to distinguish missense mutations that may cause functional changes in proteins from those that probably would not.

We used two pieces of information to analyze these mutations: (i) the location of residues in the structure of Lac tetramer bound to DNA and (ii) the log-odds scores in a position-specific scoring matrix (PSSM) generated from a multiple sequence alignment of repressor sequences. The PSSM includes two pieces of information: whether a particular site is well conserved in proteins related to Lac repressor and whether a proposed mutation is very different in physical character to residues at that position in proteins related to Lac repressor.

We defined four categories for location of an amino acid: face, buried, edge or surface depending on the value of the relative surface accessibility of a side-chain in the Lac repressor tetramer/DNA structure. Face residues were those that had reduced accessibility of their side-chains in the full complex compared to the Lac repressor monomer alone. Buried residues had less than 5% surface accessibility of their side-chains in the monomers. Edge residues were those with 5–30% accessibility and surface were all others. Surface residues are therefore those that are both on the surface and not in any binding site. The results are shown in Figure 11.

The data indicate that dissimilarity to amino acids in the repressor family as well as a location either buried in the hydrophobic core or in interfaces is sufficient to distinguish levels of "risk", i.e. a mutation with functional consequences. For each category of PSSM log-odds except PSSM = –4, face mutations are more likely to be deleterious than buried mutations and these are much more likely to be deleterious than edge or surface mutations. Interface residues do not tolerate even conservative mutations, so that even for PSSM values of 1 and 2, the proportion of deleterious mutations is 30% or higher. Mutations on the surface but not in an interface are very tolerant to mutations, with less than 1% of 743 mutations on the surface with PSSM of –2 or above having a negative phenotype.

The data in Figure 11 indicate the importance of modeling the biological unit, since all of the mutations in the "face" region would be considered surface or edge residues if the monomer was not present in a dimer complex with DNA.

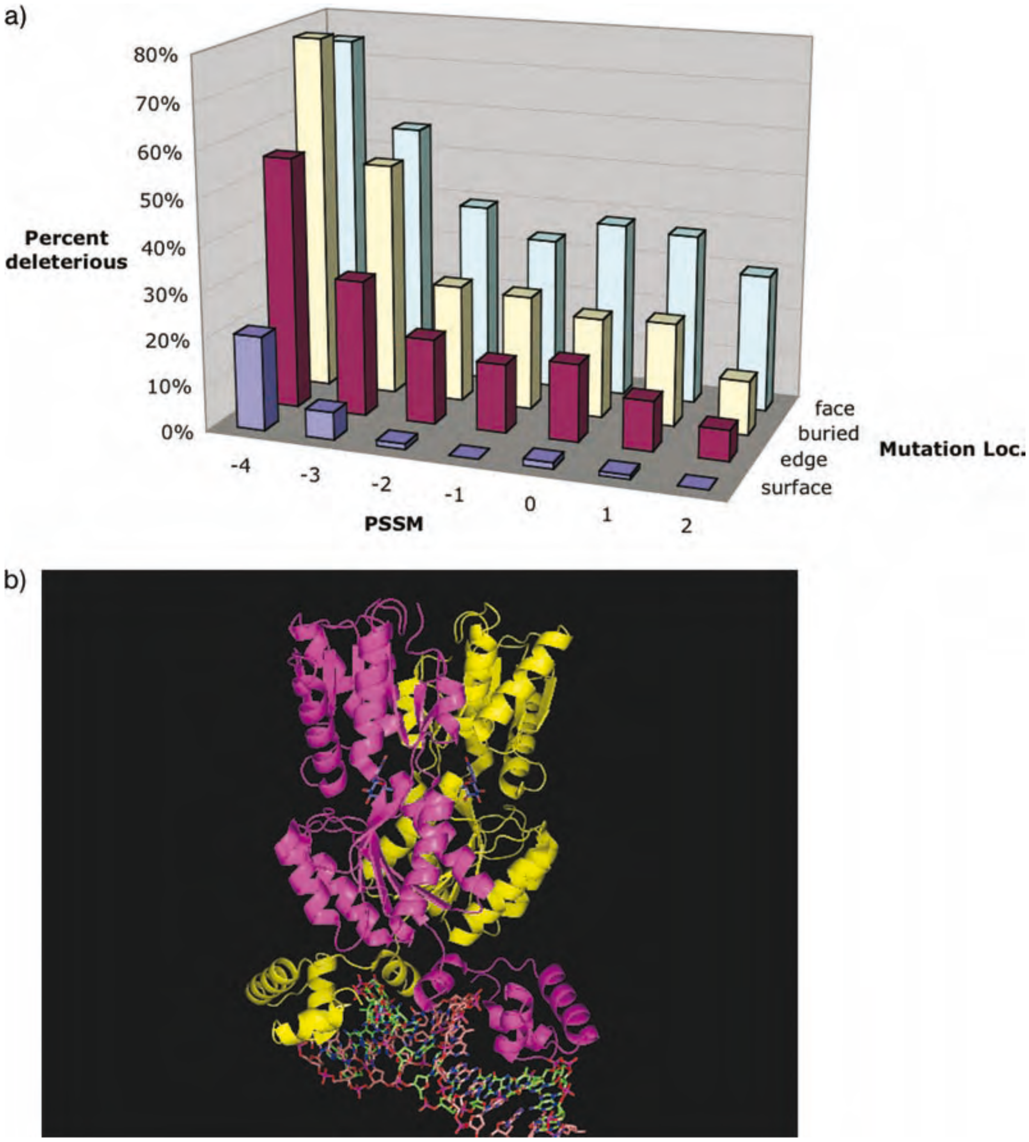


Figure 11 (a) Rates of deleterious mutations based on PSSM score and physical location for 4042 Lac repressor mutants. Face residues are those with lower surface accessibility in the Lac repressor dimer/DNA complex than in the Lac repressor monomer structure. Buried residues have less than 5% surface accessibility in the Lac repressor monomer. Surface residues have greater than 30% accessibility in the Lac repressor monomer or dimer/DNA complex, and "edge" residues have between 5 and 30% accessibility. PSSM is the log-odds score of finding the mutation at each location of the Lac repressor sequence based on a multiple alignment of homologous sequences. (b) Crystal structure of the Lac repressor dimer bound to DNA.

5 Strengths and Limitations

The strengths of homology modeling are based on the insights provided for protein function, structure and evolution that would not be available in the absence of an experimental structure. In many situations, a model built by homology is sufficient for interpreting a great deal of experimental information and will provide enough information for designing new experiments. Homology modeling may also provide functional information beyond the identification of homologous sequences to the target, i.e. a model may serve to distinguish orthologous and paralogous relationships.

The limitations are due to decreasing accuracy as the evolutionary distance between target and template increases. Alignment becomes more uncertain, insertions and deletions more frequent, and even secondary structural units may be of different lengths, numbers, and positions in very remote homologs. Predicting the locations of secondary structure units that are not present in the template structure is a difficult problem and there has been little attention paid to this problem.

The limitations of homology modeling also arise when we have insufficient information to build a model for an entire protein. For instance, we may be able to model one or more domains of a multi-domain protein or a multisubunit complex, but it may not be possible to predict the relative organization of the domains or subunits within the full protein. This remains a challenge for further research. And we are of course limited by structures present in the PDB, which are almost exclusively soluble proteins. Up to 30% of some genomes are membrane proteins, which are at present difficult to model because of the small number of membrane proteins of known structure. Recent structures [119, 146] of the G-protein-coupled receptor (GPCR) rhodopsin at higher resolution than previous structures [151] create new opportunities to model many of these membrane proteins more accurately. A number of GPCRs have been modeled on the bovine rhodopsin structures [33, 58, 136, 144, 152, 243]. In addition, recent structures of bacterial ATP-binding cassette (ABC) transporters at various stages of the transport process [45, 122, 164, 186] also provide opportunities for modeling of a large number of human ABC transporters implicated in drug resistance, such as P-glycoprotein and the multidrug resistance protein (MRP) proteins [160, 190].

Another problem is the quality of data in sequencing and structure determination. There are substantial errors in determining protein sequences from genome sequences, either because of errors in the DNA sequence or in locating exons in eukaryotic genomic DNA [199]. Over 50% of X-ray structures are solved at relatively low resolution, levels of greater than 2.0 Å. Despite progress in determining protein structures by NMR, these structures are of lower resolution than high-quality X-ray structures. While high-throughput structure determination will be of great value to modeling by homology, one concern is the quality of structure determination when the function of the proteins being determined is unknown.

6 Validation

Validation for homology modeling is available in two distinct ways: (i) the prediction rates for each method based on the prediction of known structures given information from other structures and (ii) criteria used to judge each model individually. Most structure prediction method papers have included predictions of known structure, serving as test sets of their accuracy. However, in many cases the number of test cases is inadequate (see Ref. [48]). It is also very easy to select test structures that behave particularly well for a given method and many methods do not stand up to scrutiny of large test sets performed by other researchers. Test sets vary in number of test cases as well as whether predictions of loops or side-chains are performed by building replacements on the template structure scaffold, or in real homology modeling situations where the loops/side-chains are built on nonself scaffolds. The realistic case is more difficult to perform in a comprehensive way, since it requires many sequence–structure alignments to provide the input information on which models are to be built. Another problem is that each method is judged using widely varying criteria, and so no head-to-head comparison is possible from the published papers. The problem of biased test sets and subsequent development of larger benchmarks has a long history in the secondary structure prediction field [170, 239].

While sequence alignment methods have been extensively benchmarked [184], programs that build coordinates from alignments, including the backbone, loops and side chains, have not been extensively compared to one another in large-scale tests. Recently, however, Wallner and Elofsson [220] compared several programs that build coordinates from templates given template-target alignments, including MODELLER [174], SegMod/ENCAD [116], SWISS-MODEL [189], 3D-JIGSAW [11], nest [234, 235], Builder [96, 97] and SCWRL (for side chains without modification of the template backbone) [30]. They found that three of the programs, MODELLER, nest and SegMod/ENCAD, perform better than the others. In particular, SegMod is a very old program and still performs as well as much more recent programs. They also observed that none of the homology modeling programs builds side chains as well as SCWRL.

6.1 The CASP Meeting

Another forum for testing homology modeling methods has been the ongoing series of CASP meetings organized by John Moult and colleagues [138–141, 214,215,240]. In the spring and summer before each meeting held in December 1994, 1996, 1998, 2000, 2002 and 2004, sequences of proteins whose structure was under active experimental determination by NMR or X-ray crystallography were distributed via the Internet. Anyone can submit structure predictions at various levels of detail (secondary structure predictions, sequence alignments to structures and full 3-D coordinates) before specific expiration dates for each target sequence. The models are evaluated via a number of computer programs written for the purpose, and then assessed by experts in each field, including comparative modeling, fold recognition and *ab initio* structure prediction. The organizers then invite predictors whose predictions are outstanding to present their methods and results at the meeting, and to describe their work in a special issue of the journal *Proteins*, published in the following year.

Ordinarily when protein structure prediction methods are developed, they are tested on sets of protein structures where the answer is known. Unfortunately, it is easy to select targets, even subconsciously, for which a particular method under development may work well. Also, it is easy to optimize parameters for a small test set that do not work as well for larger test sets. While the number of prediction targets in CASP is limited to numbers on the order of 10–20 per category, these numbers are still higher than many of the test sets used in testing new methods under development.

6.2 Protein Health

A number of programs have been developed to ascertain the quality of experimentally determined structures and these can be used to determine whether a protein model obeys appropriate stereochemical rules. The two most popular programs are ProCheck and WhatCheck [219]. Recently, the Richardson group has developed MolProbity, which seeks to identify a number of features in protein structures that are statistically unlikely, when compared to a manually curated set of very high-resolution structures [39]. This is a webserver that reports bad rotamer conformations, close contacts, flipped amide side-chains and other potential errors in structures. Although this site is more geared to analysis of new experimental structures, it can also be used on homology models to identify steric clashes or poorly modeled regions of proteins.

These programs check bond lengths and angles, dihedral angles, planarity of sp^2 groups, nonbonded atomic distances, disulfide bonds and other characteristics of protein structures. One of the more useful checks is to see whether backbone geometries are in acceptable regions of the Ramachandran map. Backbone conformations in the forbidden regions are very likely to be incorrect. It should be noted once again that correct geometry is no guarantee of correct structure prediction. In some cases, it may be better to tolerate a few steric conflicts or bad dihedral angles, rather than to minimize the structure's energy. While the geometry may look better, the final structure may be further away from the true structure (if it were known) than the unminimized structure. Chapter 11 discusses the problem of health of protein models and describes the respective Model Quality Assessment Programs (MQAPs).

Acknowledgements

This work was funded in part by grants R01-HG02302 and CA06927 from the National Institutes of Health. I thank Guoli Wang for providing the genome analysis in Figure 2, and Adrian Canutescu for assistance with MolIDE and Figure 3.

References

1 Alexandrov, N. N. and R. Luethy. 1998. Alignment algorithm for homology modeling and threading. Protein Sci. **7**: 254–8.

2 Allorge, D., D. Breant, J. Harlow, *et al.* 2005. Functional analysis of CYP2D6.31 variant: homology modeling suggests possible disruption of redox partner interaction by Arg440His substitution. Proteins **59**: 339–46.

3 Altschul, S. F., T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman. 1997. Gapped BLAST and PSI-BLAST: a new generation of database programs. Nucleic Acids Res. **25**: 3389–402.

4 Anderson, R. J., Z. Weng, R. K. Campbell and X. Jiang. 2005. Main-chain conformational tendencies of amino acids. Proteins **60**: 697–89.

5 Andreeva, A., D. Howorth, S. E. Brenner, T. J. Hubbard, C. Chothia and A. G. Murzin. 2004. SCOP database in 2004: refinements integrate structure and sequence family data. Nucleic Acids Res. **32**: D226–9.

6 Bailey, T. L. and M. Gribskov. 1996. The megaprior heuristic for discovering protein sequence patterns. Proc. ISMB **4**: 15–24.

7 Bairoch, A., R. Apweiler, C. H. Wu, *et al.* 2005. The Universal Protein Resource (UniProt). Nucleic Acids Res. **33**: D154–9.

8 Baldi, P., Y. Chauvin, T. Hunkapiller and M. A. McClure. 1994. Hidden Markov models of biological primary sequence information. Proc. Natl Acad. Sci. USA **91**: 1059–63.

9 Barre, A., J. P. Borges, R. Culerrier and P. Rouge. 2005. Homology modelling of the major peanut allergen Ara h 2 and surface mapping of IgE-binding epitopes. Immunol Lett. **100**: 153–8.

10 Bates, P. A., R. M. Jackson and M. J. Sternberg. 1997. Model building by comparison: a combination of expert knowledge and computer automation. Proteins **Suppl. 1**: 59–67.

11 Bates, P. A., L. A. Kelley, R. M. MacCallum and M. J. Sternberg. 2001. Enhancement of protein modeling by human intervention in applying the automatic programs 3D-JIGSAW and 3D-PSSM. Proteins **Suppl. 5**: 39–46.

12 Benedetti, E., G. Morelli, G. Nemethy and H. A. Scheraga. 1983. Statistical and energetic analysis of sidechain conformations in oligopeptides. Int. J. Peptide Protein Res. **22**: 1–15.

13 Benson, D. A., I. Karsch-Mizrachi, D. J. Lipman, J. Ostell and D. L. Wheeler. 2005. GenBank. Nucleic Acids Res. **33**: D34–8.

14 Berman, H. M., J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne. 2000. The Protein Data Bank. Nucleic Acids Res. **28**: 235–42.

15 Berman, H. M. and J. D. Westbrook. 2004. The impact of structural genomics on the protein data bank. Am. J. Pharmacogenomics **4**: 247–52.

16 Bertaccini, E. J., J. Shapiro, D. L. Brutlag and J. R. Trudell. 2005. Homology modeling of a human glycine alpha 1 receptor reveals a plausible anesthetic binding site. J. Chem. Inf. Model. **45**: 128–35.

17 Bhat, T. N., V. Sasisekharan and M. Vijayan. 1979. An analysis of sidechain conformation in proteins. Int. J. Peptide Protein Res. **13**: 170–84.

18 Blundell, T. L., B. L. Sibanda, M. J. E. Sternberg and J. M. Thornton. 1987. Knowledge-based prediction of protein structures and the design of novel molecules. Nature **326**: 347–52.

19 Boomsma, W. and T. Hamelryck. 2005. Full cyclic coordinate descent: solving the protein loop closure problem in Calpha space. BMC Bioinformatics **6**: 159.

20 Borysenko, C. W., W. F. Furey and H. C. Blair. 2005. Comparative modeling of TNFRSF25 (DR3) predicts receptor destabilization by a mutation linked to rheumatoid arthritis. Biochem. Biophys. Res. Commun. **328**: 794–9.

21 Boudeau, J., J. W. Scott, N. Resta, *et al.* 2004. Analysis of the LKB1–STRAD–MO25 complex. J. Cell Sci. **117**: 6365–75.

22 Bower, M. J., F. E. Cohen and R. L. Dunbrack, Jr. 1997. Prediction of protein side-chain rotamers from a backbone-dependent rotamer library: a new homology modeling tool. J. Mol. Biol. **267**: 1268–82.

23 Brown, M., R. Hughey, A. Krogh, I. S. Mian, K. Sjolander and D. Haussler. 1993. Using Dirichlet mixture priors to derive hidden Markov models for protein families. Proc. ISMB **1**: 47–55.

24 Browne, W. J., A. C. North and D. C. Phillips. 1969. A possible three-dimensional structure of bovine alpha-lactalbumin based on that of hen's egg-white lysozyme. J. Mol. Biol. **42**: 65–86.

25 Bruccoleri, R. E. and M. Karplus. 1987. Prediction of the folding of short polypeptide segments by uniform conformational sampling. Biopolymers **26**: 137–68.

26 Caffrey, C. R., L. Placha, C. Barinka, *et al.* 2005. Homology modeling and SAR analysis of *Schistosoma japonicum* cathepsin D (SjCD) with statin inhibitors identify a unique active site steric barrier with potential for the design of specific inhibitors. Biol. Chem. **386**: 339–49.

27 Campillo, N. E., J. Antonio Paez, L. Lagartera and A. Gonzalez. 2005. Homology modelling and active-site-mutagenesis study of the catalytic domain of the pneumococcal phosphorylcholine esterase. Bioorg. Med. Chem. **13**: 6404–13.

28 Canutescu, A. A. and R. L. Dunbrack, Jr. 2003. Cyclic coordinate descent: a robotics algorithm for protein loop closure. Protein Sci. **12**: 963–72.

29 Canutescu, A. A. and R. L. Dunbrack, Jr. 2005. MollDE: a homology modeling framework you can click with. Bioinformatics **21**: 2914–6.

30 Canutescu, A. A., A. A. Shelenkov and R. L. Dunbrack, Jr. 2003. A graph-theory algorithm for rapid protein side-chain prediction. Protein Sci. **12**: 2001–14.

31 Chothia, C. and A. M. Lesk. 1986. The relation between the divergence of sequence and structure in proteins. EMBO J. **5**: 823–6.

32 Chothia, C., A. M. Lesk, A. Tramontano, *et al.* 1989. Conformations of immunoglobulin hypervariable regions. Nature **342**: 877–83.

33 COSTANZI, S., L. MAMEDOVA, Z. G. GAO AND K. A. JACOBSON. 2004. Architecture of P2Y nucleotide receptors: structural comparison based on sequence analysis, mutagenesis, and homology modeling. J. Med. Chem. **47**: 5393–404.

34 COUCH, F. J. AND B. L. WEBER. 1996. Mutations and polymorphisms in the familial early-onset breast cancer (BRCA1) gene. Breast Cancer Information Core. Hum. Mutat. **8**: 8–18.

35 COUTSIAS, E. A., C. SEOK, M. P. JACOBSON AND K. A. DILL. 2004. A kinematic view of loop closure. J. Comput. Chem. **25**: 510–28.

36 DA SILVA, C. H., I. CARVALHO AND C. A. TAFT. 2005. Homology modeling and molecular interaction field studies of alpha-glucosidases as a guide to structure-based design of novel proposed anti-HIV inhibitors. J. Comput. Aided Mol. Des. **19**: 83–92.

37 DAHIYAT, B. I., C. A. SARISKY AND S. L. MAYO. 1997. De novo protein design: towards fully automated sequence selection. J. Mol. Biol. **273**: 789–96.

38 DASGUPTA, B., L. PAL, G. BASU AND P. CHAKRABARTI. 2004. Expanded turn conformations: characterization and sequence–structure correspondence in alpha-turns with implications in helix folding. Proteins **55**: 305–15.

39 DAVIS, I. W., L. W. MURRAY, J. S. RICHARDSON AND D. C. RICHARDSON. 2004. MOLPROBITY: structure validation and all-atom contact analysis for nucleic acids and their complexes. Nucleic Acids Res. **32**: W615–9.

40 DE MAEYER, M., J. DESMET AND I. LASTERS. 2000. The dead-end elimination theorem: mathematical aspects, implementation, optimizations, evaluation, and performance. Methods Mol. Biol. **143**: 265–304.

41 DESMET, J., M. DE MAEYER, B. HAZES AND I. LASTERS. 1992. The dead-end elimination theorem and its use in protein sidechain positioning. Nature **356**: 539–42.

42 DESMET, J., M. DE MAEYER AND I. LASTERS. 1997. Theoretical and algorithmical optimization of the dead-end elimination theorem. Pac. Symp. Biocomput. 1997: 122–33.

43 DODGE, C., R. SCHNEIDER AND C. SANDER. 1998. The HSSP database of protein structure–sequence alignments and family profiles. Nucleic Acids Res. **26**: 313–5.

44 DONATE, L. E., S. D. RUFINO, L. H. J. CANARD AND T. L. BLUNDELL. 1996. Conformational analysis and clustering of short and medium size loops connecting regular secondary structures: a database for modeling and prediction. Protein Sci. **5**: 2600–16.

45 DONG, J., G. YANG AND H. S. MCHAOURAB. 2005. Structural basis of energy transduction in the transport cycle of MsbA. Science **308**: 1023–8.

46 DORIT, R. L., L. SCHOENBACH AND W. GILBERT. 1990. How big is the universe of exons? Science **250**: 1377–82.

47 DUNBRACK, R. L., JR. 1999. Comparative modeling of CASP3 targets using PSI-BLAST and SCWRL. Proteins **Suppl. 3**: 81–7.

48 DUNBRACK, R. L., JR. 2001. Protein structure prediction in biology and medicine. In LENGAUER, T. (ed.), *Bioinformatics and Drug Design*. Wiley, New York, NY: 145–235.

49 DUNBRACK, R. L., JR. 2002. Rotamer libraries in the 21st century. Curr. Opin. Struct. Biol. **12**: 431–40.

50 DUNBRACK, R. L., JR. AND F. E. COHEN. 1997. Bayesian statistical analysis of protein sidechain rotamer preferences. Protein Sci. **6**: 1661–81.

51 DUNBRACK, R. L., JR. AND M. KARPLUS. 1993. Backbone-dependent rotamer library for proteins. Application to side-chain prediction. J. Mol. Biol. **230**: 543–74.

52 DUNBRACK, R. L., JR. AND M. KARPLUS. 1994. Conformational analysis of the backbone-dependent rotamer preferences of protein sidechains. Nat. Struct. Biol. **1**: 334–40.

53 EDDY, S. R. 1996. Hidden Markov models. Curr. Opin. Struct. Biol. **6**: 361–5.

54 EDDY, S. R., G. MITCHISON AND R. DURBIN. 1995. Maximum discrimination hidden Markov models of sequence consensus. J. Comput. Biol. **2**: 9–23.

55 Edgar, R. C. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. **32**: 1792–7.

56 Espadaler, J., N. Fernandez-Fuentes, A. Hermoso, E. Querol, F. X. Aviles, M. J. Sternberg and B. Oliva. 2004. ArchDB: automated protein loop classification as a tool for structural genomics. Nucleic Acids Res. **32**: D185–8.

57 Evans, J. S., S. I. Chan and W. A. Goddard, 3rd. 1995. Prediction of polyelectrolyte polypeptide structures using Monte Carlo conformational search methods with implicit solvation modeling. Protein Sci. **4**: 2019–31.

58 Evers, A. and T. Klabunde. 2005. Structure-based drug discovery using GPCR homology modeling: successful virtual screening for antagonists of the alpha1A adrenergic receptor. J. Med. Chem. **48**: 1088–97.

59 Fan, H. and A. E. Mark. 2004. Refinement of homology-based protein structures by molecular dynamics simulation techniques. Protein Sci. **13**: 211–20.

60 Fechteler, T., U. Dengler and D. Schomburg. 1995. Prediction of protein three-dimensional structures in insertion and deletion regions: a procedure for searching data bases of representative protein fragments using geometric scoring criteria. J. Mol. Biol. **253**: 114–31.

61 Fernandez-Fuentes, N., E. Querol, F. X. Aviles, M. J. Sternberg and B. Oliva. 2005. Prediction of the conformation and geometry of loops in globular proteins: testing ArchDB, a structural classification of loops. Proteins **60**: 746–57.

62 Fetrow, J. S., A. Godzik and J. Skolnick. 1998. Functional analysis of the *Escherichia coli* genome using the sequence-to-structure-to-function paradigm: identification of proteins exhibiting the glutaredoxin/thioredoxin disulfide oxidoreductase activity. J. Mol. Biol. **282**: 703–11.

63 Fidelis, K., P. S. Stern, D. Bacon and J. Moult. 1994. Comparison of systematic search and database methods for constructing segments of protein structures. Protein Eng. **7**: 953–60.

64 Fine, R. M., H. Wang, P. S. Shenkin, D. L. Yarmush and C. Levinthal. 1986. Predicting antibody hypervariable loop conformations. II: minimization and molecular dynamics studies of MCPC603 from many randomly generated loop conformations. Proteins **1**: 342–62.

65 Fiser, A., R. K. Do and A. Sali. 2000. Modeling of loops in protein structures. Protein Sci. **9**: 1753–73.

66 Fogolari, F. and S. C. Tosatto. 2005. Application of MM/PBSA colony free energy to loop decoy discrimination: toward correlation between energy and root mean square deviation. Protein Sci. **14**: 889–901.

67 Franca, T. C., P. G. Pascutti, T. C. Ramalho and J. D. Figueroa-Villar. 2005. A three-dimensional structure of *Plasmodium falciparum* serine hydroxymethyltransferase in complex with glycine and 5-formyl-tetrahydrofolate. Homology modeling and molecular dynamics. Biophys. Chem. **115**: 1–10.

68 Gallicchio, E., L. Y. Zhang and R. M. Levy. 2002. The SGB/NP hydration free energy model based on the surface generalized born solvent reaction field and novel nonpolar hydration free energy estimators. J. Comput. Chem. **23**: 517–29.

69 Geetha, V. and P. J. Munson. 1997. Linkers of secondary structures in proteins. Protein Sci. **6**: 2538–47.

70 Goldstein, R. F. 1994. Efficient rotamer elimination applied to protein side-chains and related spin glasses. Biophys. J. **66**: 1335–40.

71 Gordon, D. B. and S. L. Mayo. 1999. Branch-and-terminate: a combinatorial optimization algorithm for protein design. Struct. Fold. Des. **7**: 1089–98.

72 Greer, J. 1990. Comparative modeling methods: application to the family of the mammalian serine proteases. Proteins **7**: 317–34.

73 Greer, J. 1980. Model for haptoglobin heavy chain based upon structural homology. Proc. Natl Acad. Sci. USA **77**: 3393–7.

74 Guex, N. and M. C. Peitsch. 1997. SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling. Electrophoresis **18**: 2714–23.

75 Gunsteren, W. F. v., P. H. Hünenberger, A. E. Mark, P. E. Smith and I. G. Tironi. 1995. Computer simulation of protein motion. Comp. Phys. Commun. **91**: 305–19.

76 Hamosh, A., A. F. Scott, J. Amberger, D. Valle and V. A. McKusick. 2000. Online Mendelian Inheritance in Man (OMIM). Hum. Mutat. **15**: 57–61.

77 Hamza, A., H. Cho, H. H. Tai and C. G. Zhan. 2005. Understanding human 15-hydroxyprostaglandin dehydrogenase binding with NAD^+ and PGE_2 by homology modeling, docking and molecular dynamics simulation. Bioorg. Med. Chem. **13**: 4544–51.

78 Havel, T. F. and M. E. Snow. 1991. A new method for building protein conformations from sequence alignments with homologues of known structure. J. Mol. Biol. **217**: 1–7.

79 Hemminki, A., D. Markie, I. Tomlinson, *et al.* 1998. A serine/threonine kinase gene defective in Peutz–Jeghers syndrome. Nature **391**: 184–7.

80 Henrick, K. and J. M. Thornton. 1998. PQS: a protein quaternary structure file server. Trends Biochem. Sci. **23**: 358–61.

81 Heuser, P., G. Wohlfahrt and D. Schomburg. 2004. Efficient methods for filtering and ranking fragments for the prediction of structurally variable regions in proteins. Proteins **54**: 583–95.

82 Holm, L. and C. Sander. 1995. Dali: a network tool for protein structure comparison. Trends Biochem. Sci. **20**: 478–80.

83 Holm, L. and C. Sander. 1992. Fast and simple Monte Carlo algorithm for side chain optimization in proteins: application to model building by homology. Proteins **14**: 213–23.

84 Holm, L. and C. Sander. 1996. Mapping the protein universe. Science **273**: 595–603.

85 Huynen, M., T. Doerks, F. Eisenhaber, C. Orengo, S. Sunyaev, Y. Yuan and P. Bork. 1998. Homology-based fold predictions for *Mycoplasma genitalium* proteins. J. Mol. Biol. **280**: 323–6.

86 Hwang, J. K. and W. F. Liao. 1995. Side-chain prediction by neural networks and simulated annealing optimization. Protein Eng. **8**: 363–70.

87 Jacobson, M. P., D. L. Pincus, C. S. Rapp, T. J. Day, B. Honig, D. E. Shaw and R. A. Friesner. 2004. A hierarchical approach to all-atom protein loop prediction. Proteins **55**: 351–67.

88 Janin, J., S. Wodak, M. Levitt and B. Maigret. 1978. Conformations of amino acid side-chains in proteins. J. Mol. Biol. **125**: 357–86.

89 Jenne, D. E., H. Reimann, J. Nezu, *et al.* 1998. Peutz–Jeghers syndrome is caused by mutations in a novel serine threonine kinase. Nat. Genet. **18**: 38–43.

90 Jones, D. T. 1999. Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. **292**: 195–202.

91 Jones, T. A. and S. Thirup. 1986. Using known substructures in protein model building and crystallography. EMBO J. **5**: 819–22.

92 Karchin, R., M. Diekhans, L. Kelly, D. J. Thomas, U. Pieper, N. Eswar, D. Haussler and A. Sali. 2005. LS-SNP: large-scale annotation of coding non-synonymous SNPs based on multiple information sources. Bioinformatics **21**: 2814–20.

93 Karplus, K., C. Barrett and R. Hughey. 1998. Hidden Markov models for detecting remote protein homologies. Bioinformatics **14**: 846–56.

94 Karplus, M. and R. G. Parr. 1963. An approach to the internal rotation problem. J. Chem. Phys. **38**: 1547–52.

95 Kim, S. H., D. H. Shin, I. G. Choi, U. Schulze-Gahmen, S. Chen and R. Kim. 2003. Structure-based functional inference in structural genomics. J. Struct. Funct. Genomics **4**: 129–35.

96 Koehl, P. and M. Delarue. 1994. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. J. Mol. Biol. **239**: 249–75.

97 Koehl, P. and M. Delarue. 1995. A self consistent mean field approach to simultaneous gap closure and side-chain positioning in homology modelling. Nat. Struct. Biol. **2**: 163–70.

98 Koehl, P. and M. Levitt. 1999. A brighter future for protein structure prediction. Nat. Struct. Biol. **6**: 108–11.

99 Kono, H. and J. Doi. 1994. Energy minimization method using automata network for sequence and side-chain conformation prediction from given backbone geometry. Proteins **19**: 244–55.

100 Kopp, J. and T. Schwede. 2004. The SWISS-MODEL Repository of annotated three-dimensional protein structure homology models. Nucleic Acids Res. **32**: D230–4.

101 Kraus, J. P., M. Janosik, V. Kozich, *et al.* 1999. Cystathionine beta-synthase mutations in homocystinuria. Hum. Mutat. **13**: 362–75.

102 Krawczak, M., E. V. Ball, I. Fenton, P. D. Stenson, S. Abeysinghe, N. Thomas and D. N. Cooper. 2000. Human gene mutation database – a biomedical information and research resource. Hum. Mutat. **15**: 45–51.

103 Kumar, M., M. Bhasin, N. K. Natt and G. P. Raghava. 2005. BhairPred: prediction of beta-hairpins in a protein from multiple alignment information using ANN and SVM techniques. Nucleic Acids Res. **33**: W154–9.

104 Kwasigroch, J., J. Chomilier and J. Mornon. 1996. A global taxonomy of loops in globular proteins. J. Mol. Biol. **259**: 855–72.

105 Lambert, C., N. Leonard, X. De Bolle and E. Depiereux. 2002. ESyPred3D: prediction of proteins 3D structures. Bioinformatics **18**: 1250–6.

106 Laskowski, R. A., J. D. Watson and J. M. Thornton. 2005. ProFunc: a server for predicting protein function from 3D structure. Nucleic Acids Res. **33**: W89–93.

107 Lasters, I., M. De Maeyer and J. Desmet. 1995. Enhanced dead-end elimination in the search for the global minimum energy conformation of a collection of protein side chains. Protein Eng. **8**: 815–22.

108 Lasters, I. and J. Desmet. 1993. The fuzzy-end elimination theorem: correctly implementing the sidechain placement algorithm based on the dead-end elimination theorem. Protein Eng. **6**: 717–22.

109 Laughton, C. A. 1994. Prediction of protein sidechain conformations from local three-dimensional homology relationships. J. Mol. Biol. **235**: 1088–97.

110 Launonen, V. 2005. Mutations in the human LKB1/STK11 gene. Hum. Mutat. **26**: 291–7.

111 Leaver-Fay, A., B. Kuhlman and J. Snoeyink. 2005. An adaptive dynamic programming algorithm for the side chain placement problem. Pac. Symp. Biocomput.: 16–27.

112 Lee, C. and S. Subbiah. 1991. Prediction of protein side-chain conformation by packing optimization. J. Mol. Biol. **217**: 373–88.

113 Lessel, U. and D. Schomburg. 1997. Creation and characterization of a new, non-redundant fragment data bank. Protein Eng. **10**: 659–64.

114 Lessel, U. and D. Schomburg. 1999. Importance of anchor group positioning in protein loop prediction. Proteins **37**: 56–64.

115 Leszczynski, J. F. and G. D. Rose. 1986. Loops in globular proteins: a novel category of secondary structure. Science **234**: 849–55.

116 Levitt, M. 1992. Accurate modeling of protein conformation by automatic segment matching. J. Mol. Biol. **226**: 507–33.

117 Levitt, M., M. Gerstein, E. Huang, S. Subbiah and J. Tsai. 1997. Protein folding: the endgame. Annu. Rev. Biochem. **66**: 549–79.

118 Li, H., R. Tejero, D. Monleon, D. Bassolino-Klimas, C. Abate-Shen, R. E. Bruccoleri and G. T. Montelione. 1997. Homology modeling using simulated annealing of restrained molecular dynamics and conformational search calculations with CONGEN: application in predicting the three-dimensional structure of murine homeodomain Msx-1. Protein Sci. **6**: 956–70.

119 LI, J., P. C. EDWARDS, M. BURGHAMMER, C. VILLA AND G. F. SCHERTLER. 2004. Structure of bovine rhodopsin in a trigonal crystal form. J. Mol. Biol. **343**: 1409–38.

120 LIANG, S. AND N. V. GRISHIN. 2002. Side-chain modeling with an optimized scoring function. Protein Sci. **11**: 322–31.

121 LICHTARGE, O., H. R. BOURNE AND F. E. COHEN. 1996. An evolutionary trace method defines binding surfaces common to protein families. J. Mol. Biol. **257**: 342–58.

122 LOCHER, K. P., A. T. LEE AND D. C. REES. 2002. The *E. coli* BtuCD structure: a framework for ABC transporter architecture and mechanism. Science **296**: 1091–8.

123 LOOGER, L. L. AND H. W. HELLINGA. 2001. Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: implications for protein design and structural genomics. J. Mol. Biol. **307**: 429–45.

124 LOVELL, S. C., I. W. DAVIS, W. B. ARENDALL, 3RD, P. I. DE BAKKER, J. M. WORD, M. G. PRISANT, J. S. RICHARDSON AND D. C. RICHARDSON. 2003. Structure validation by Calpha geometry: phi,psi and Cbeta deviation. Proteins **50**: 437–50.

125 LOVELL, S. C., J. M. WORD, J. S. RICHARDSON AND D. C. RICHARDSON. 2000. The penultimate rotamer library. Proteins **40**: 389–408.

126 LUNDSTROM, K. 2004. Structural genomics on membrane proteins: mini review. Comb. Chem. High Throughput Screen. **7**: 431–9.

127 MACKERELL, A. D., JR., D. BASHFORD, M. BELLOTT, *et al.* 1998. All-atom empirical potential for molecular modeling and dynamics studies of proteins. J. Phys. Chem. B**102**: 3586–616.

128 MARABOTTI, A. AND A. M. FACCHIANO. 2005. Homology modeling studies on human galactose-1-phosphate uridylyltransferase and on its galactosemia-related mutant Q188R provide an explanation of molecular effects of the mutation on homo- and heterodimers. J. Med. Chem. **48**: 773–9.

129 MARINELLI, L., K. E. GOTTSCHALK, A. MEYER, E. NOVELLINO AND H. KESSLER. 2004. Human integrin alphavbeta5: homology modeling and ligand binding. J. Med. Chem. **47**: 4166–77.

130 MARKIEWICZ, P., L. G. KLEINA, C. CRUZ, S. EHRET AND J. H. MILLER. 1994. Genetic studies of the lac repressor. XIV. Analysis of 4000 altered *Escherichia coli* lac repressors reveals essential and non-essential residues, as well as "spacers" which do not require a specific sequence. J. Mol. Biol. **240**: 421–33.

131 MARTIN, A. C., K. TODA, H. J. STIRK AND J. M. THORNTON. 1995. Long loops in proteins. Protein Eng. **8**: 1093–101.

132 MCGREGOR, M. J., S. A. ISLAM AND M. J. E. STERNBERG. 1987. Analysis of the relationship between sidechain conformation and secondary structure in globular proteins. J. Mol. Biol. **198**: 295–310.

133 MENDES, J., A. M. BAPTISTA, M. A. CARRONDO AND C. M. SOARES. 1999. Improved modeling of side-chains in proteins with rotamer-based methods: a flexible rotamer model. Proteins **37**: 530–43.

134 MENDES, J., C. M. SOARES AND M. A. CARRONDO. 1999. Improvement of side-chain modeling in proteins with the self-consistent mean field theory method based on an analysis of the factors influencing prediction. Biopolymers **50**: 111–31.

135 MICHALSKY, E., A. GOEDE AND R. PREISSNER. 2003. Loops In Proteins (LIP) – a comprehensive loop database for homology modelling. Protein Eng. **16**: 979–85.

136 MIEDLICH, S. U., L. GAMA, K. SEUWEN, R. M. WOLF AND G. E. BREITWIESER. 2004. Homology modeling of the transmembrane domain of the human calcium sensing receptor and localization of an allosteric binding site. J. Biol. Chem. **279**: 7254–63.

137 MISURA, K. M., A. V. MOROZOV AND D. BAKER. 2004. Analysis of anisotropic side-chain packing in proteins and application to high-resolution structure prediction. J. Mol. Biol. **342**: 651–64.

138 MOULT, J. 1996. The current state of the art in protein structure prediction. Curr. Opin. Biotechnol. **7**: 422–27.

139 MOULT, J. 1999. Predicting protein three-dimensional structure. Curr. Opin. Biotechnol. **10**: 583–8.

140 MOULT, J., T. HUBBARD, S. H. BRYANT, K. FIDELIS AND J. T. PEDERSEN. 1997. Critical assessment of methods of protein structure prediction (CASP): round II. Proteins **Suppl. 1**: 2–6.

141 MOULT, J., T. HUBBARD, K. FIDELIS AND J. T. PEDERSEN. 1999. Critical assessment of methods of protein structure prediction (CASP): round III. Proteins **Suppl. 3**: 2–6.

142 NAJMANOVICH, R. J., J. W. TORRANCE AND J. M. THORNTON. 2005. Prediction of protein function from structure: insights from methods for the detection of local structural similarities. Biotechniques **38**: 847, 849, 851.

143 NOWAKOWSKI, J., C. N. CRONIN, D. E. MCREE, *et al.* 2002. Structures of the cancer-related Aurora-A, FAK, and EphA2 protein kinases from nanovolume crystallography. Structure (Camb.) **10**: 1659–67.

144 NUNEZ MIGUEL, R., J. SANDERS, J. JEFFREYS, *et al.* 2004. Analysis of the thyrotropin receptor-thyrotropin interaction by comparative modeling. Thyroid **14**: 991–1011.

145 O'CONNELL, N. M., R. E. SAUNDERS, C. A. LEE, D. J. PERRY AND S. J. PERKINS. 2005. Structural interpretation of 42 mutations causing factor XI deficiency using homology modeling. J. Thromb. Haemost. **3**: 127–38.

146 OKADA, T., M. SUGIHARA, A. N. BONDAR, M. ELSTNER, P. ENTEL AND V. BUSS. 2004. The retinal conformation and its environment in rhodopsin in light of a new 2.2 Å crystal structure. J. Mol. Biol. **342**: 571–83.

147 OLIVA, B., P. A. BATES, E. QUEROL, F. X. AVILÉS AND M. J. E. STERNBERG. 1997. An automated classification of the structure of protein loops. J. Mol. Biol. **266**: 814–30.

148 PACE, H. C., M. A. KERCHER, P. LU, P. MARKIEWICZ, J. H. MILLER, G. CHANG AND M. LEWIS. 1997. Lac repressor genetic map in real space. Trends Biochem.Sci. **22**: 334–9.

149 PAL, L., B. DASGUPTA AND P. CHAKRABARTI. 2005. 3_{10}-Helix adjoining alpha-helix and beta-strand: sequence and structural features and their conservation. Biopolymers **78**: 147–62.

150 PAL, M. AND S. DASGUPTA. 2003. The nature of the turn in omega loops of proteins. Proteins **51**: 591–606.

151 PALCZEWSKI, K., T. KUMASAKA, T. HORI, *et al.* 2000. Crystal structure of rhodopsin: a G protein-coupled receptor. Science **289**: 739–45.

152 PEDRETTI, A., M. ELENA SILVA, L. VILLA AND G. VISTOLI. 2004. Binding site analysis of full-length alpha1a adrenergic receptor using homology modeling and molecular docking. Biochem. Biophys. Res. Commun. **319**: 493–500.

153 PEITSCH, M. C. 1997. Large scale protein modelling and model repository. Proc. ISMB **5**: 234–6.

154 PEITSCH, M. C. 1996. ProMod and Swiss-Model: Internet-based tools for automated comparative protein modelling. Biochem. Soc. Trans. **24**: 274–9.

155 PEITSCH, M. C., M. R. WILKINS, L. TONELLA, J. C. SANCHEZ, R. D. APPEL AND D. F. HOCHSTRASSER. 1997. Large-scale protein modelling and integration with the SWISS-PROT and SWISS-2DPAGE databases: the example of *Escherichia coli*. Electrophoresis **18**: 498–501.

156 PERUTZ, M. F., J. C. KENDREW AND H. C. WATSON. 1965. Structure and function of haemoglobin. J. Mol. Biol. **13**: 669–78.

157 PETERSON, R. W., P. L. DUTTON AND A. J. WAND. 2004. Improved side-chain prediction accuracy using an *ab initio* potential energy function and a very large rotamer library. Protein Sci. **13**: 735–51.

158 PIEPER, U., N. ESWAR, A. C. STUART, V. A. ILYIN AND A. SALI. 2002. MODBASE, a database of annotated comparative protein structure models. Nucleic Acids Res. **30**: 255–9.

159 PIERCE, N. A., J. A. SPRIET, J. DESMET AND S. L. MAYO. 1999. Conformational splitting: a more powerful criterion for dead-end elimination. J. Comp. Chem. **21**: 999–1009.

160 PLEBAN, K., A. MACCHIARULO, G. COSTANTINO, R. PELLICCIARI, P. CHIBA AND G. F. ECKER. 2004. Homology model of the multidrug transporter LmrA from *Lactococcus lactis*. Bioorg. Med. Chem. Lett. **14**: 5823–6.

161 PONDER, J. W. AND F. M. RICHARDS. 1987. Tertiary templates for proteins: Use of packing criteria in the enumeration of allowed sequences for different structural classes. J. Mol. Biol. **193**: 775–92.

162 PURTA, E., F. VAN VLIET, C. TRICOT, L. G. DE BIE, M. FEDER, K. SKOWRONEK, L. DROOGMANS AND J. M. BUJNICKI. 2005. Sequence–structure–function relationships of a tRNA (m7G46) methyltransferase studied by homology modeling and site-directed mutagenesis. Proteins **59**: 482–8.

163 QUEVILLON-CHERUEL, S., B. COLLINET, C. Z. ZHOU, *et al.* 2003. A structural genomics initiative on yeast proteins. J. Synchrotron Radiat. **10**: 4–8.

164 REYES, C. L. AND G. CHANG. 2005. Structure of the ABC transporter MsbA in complex with ADP·vanadate and lipopolysaccharide. Science **308**: 1028–31.

165 RING, C. S., D. G. KNELLER, R. LANGRIDGE AND F. E. COHEN. 1992. Taxonomy and conformational analysis of loops in proteins [published erratum appears in J. Mol. Biol. 1992; **227**(3): 977]. J. Mol. Biol. **224**: 685–99.

166 RING, C. S., E. SUN, J. H. MCKERROW, G. K. LEE, P. J. ROSENTHAL, I. D. KUNTZ AND F. E. COHEN. 1993. Structure-based inhibitor design by using protein models for the development of antiparasitic agents. Proc. Natl. Acad. Sci. USA **90**: 3583–7.

167 ROHL, C. A., C. E. STRAUSS, D. CHIVIAN AND D. BAKER. 2004. Modeling structurally variable regions in homologous proteins with rosetta. Proteins **55**: 656–77.

168 ROOMAN, M. J. AND S. J. WODAK. 1991. Weak correlation between predictive power of individual sequence patterns and overall prediction accuracy in proteins. Proteins **9**: 69–78.

169 ROST, B. 1999. Twilight zone of protein sequence alignments. Protein Eng. **12**: 85–94.

170 ROST, B., C. SANDER AND R. SCHNEIDER. 1994. Redefining the goals of structure prediction. J. Mol. Biol. **235**: 13–26.

171 ROY, S. AND S. SEN. 2005. Homology modeling based solution structure of Hoxc8–DNA complex: role of context bases outside TAAT stretch. J. Biomol. Struct. Dyn. **22**: 707–18.

172 RUFINO, S. D., L. E. DONATE, L. H. J. CANARD AND T. L. BLUNDELL. 1997. Predicting the conformational class of short and medium size loops connecting regular secondary structures: application to comparative modeling. J. Mol. Biol. **267**: 352–67.

173 SAHASRABUDHE, P. V., R. TEJERO, S. KITAO, Y. FURUICHI AND G. T. MONTELIONE. 1998. Homology modeling of an RNP domain from a human RNA-binding protein: homology-constrained energy optimization provides a criterion for distinguishing potential sequence alignments. Proteins **33**: 558–66.

174 SALI, A. AND T. L. BLUNDELL. 1993. Comparative protein modelling by satisfaction of spatial restraints. J. Mol. Biol. **234**: 779–815.

175 SALI, A. AND J. P. OVERINGTON. 1994. Derivation of rules for comparative protein modeling from a database of protein structure alignments. Protein Sci. **3**: 1582–96.

176 SAMUDRALA, R. AND J. MOULT. 1998. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. J. Mol. Biol. **275**: 895–916.

177 SAMUDRALA, R. AND J. MOULT. 1998. Determinants of side chain conformational preferences in protein structures. Protein Eng. **11**: 991–7.

178 Samudrala, R. and J. Moult. 1998. A graph-theoretic algorithm for comparative modeling of protein structure. J. Mol. Biol. **279**: 287–302.

179 Samudrala, R. and J. Moult. 1997. Handling context-sensitivity in protein structures using graph theory: bona fide prediction. Proteins **Suppl. 1**: 43–9.

180 Sanchez, R. and A. Sali. 1997. Evaluation of comparative protein structure modeling by MODELLER-3. Proteins **Suppl. 1**: 50–8.

181 Sanchez, R. and A. Sali. 1998. Large-scale protein structure modeling of the Saccharomyces cerevisiae genome. Proc. Natl Acad. Sci. USA **95**: 13597–602.

182 Sasisekharan, V. and P. K. Ponnuswamy. 1970. Backbone and sidechain conformations of amino acids and amino acid residues in peptides. Biopolymers **9**: 1249–56.

183 Sasisekharan, V. and P. K. Ponnuswamy. 1971. Studies on the conformation of amino acids. X. Conformations of norvalyl, leucyl, aromatic side groups in a dipeptide unit. Biopolymers **10**: 583–92.

184 Sauder, J. M., J. W. Arthur and R. L. Dunbrack, Jr. 2000. Large-scale comparison of protein sequence alignment algorithms with structure alignments. Proteins **40**: 6–22.

185 Schiffer, C. A., J. W. Caldwell, P. A. Kollman and R. M. Stroud. 1990. Prediction of homologous protein structures based on conformational searches and energetics. Proteins **8**: 30–43.

186 Schmitt, L. 2002. The first view of an ABC transporter: the X-ray crystal structure of MsbA from *E. coli*. ChemBiochem. **3**: 161–5.

187 Schrauber, H., F. Eisenhaber and P. Argos. 1993. Rotamers: To be or not to be? An analysis of amino acid sidechain conformations in globular proteins. J. Mol. Biol. **230**: 592–612.

188 Schumacher, V., T. Vogel, B. Leube, C. Driemel, T. Goecke, G. Moeslein and B. Royer-Pokora. 2005. Gene symbol: STK11. Disease: Peutz–Jeghers syndrome. Hum. Genet. **116**: 541.

189 Schwede, T., J. Kopp, N. Guex and M. C. Peitsch. 2003. SWISS-MODEL: an automated protein homology-modeling server. Nucleic Acids Res. **31**: 3381–5.

190 Seigneuret, M. and A. Garnier-Suillerot. 2003. A structural model for the open conformation of the mdr1 P-glycoprotein based on the MsbA crystal structure. J. Biol. Chem. **278**: 30115–24.

191 Shenkin, P. S., D. L. Yarmush, R. M. Fine, H. J. Wang and C. Levinthal. 1987. Predicting antibody hypervariable loop conformation. I. Ensembles of random conformations for ringlike structures. Biopolymers **26**: 2053–85.

192 Shepherd, A. J., D. Gorse and J. M. Thornton. 1999. Prediction of the location and type of beta-turns in proteins using neural networks. Protein Sci. **8**: 1045–55.

193 Shetty, R. P., P. I. De Bakker, M. A. DePristo and T. L. Blundell. 2003. Advantages of fine-grained side chain conformer libraries. Protein Eng. **16**: 963–9.

194 Shindyalov, I. N. and P. E. Bourne. 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng. **11**: 739–47.

195 Sibanda, B. L. and J. M. Thornton. 1985. Beta-hairpin families in globular proteins. Nature **316**: 170–4.

196 Sibanda, B. L. and J. M. Thornton. 1991. Conformation of beta hairpins in protein structures: classification and diversity in homologous structures. Methods Enzymol. **202**: 59–82.

197 Simons, K. T., C. Kooperberg, E. Huang and D. Baker. 1997. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J. Mol. Biol. **268**: 209–25.

198 Sowdhamini, R., S. D. Rufino and T. L. Blundell. 1996. A database of globular protein structural domains: clustering of representative family members into similar folds. Fold. Des. **1**: 209–20.

199 States, D. J. and D. Botstein. 1991. Molecular sequence accuracy and the analysis of protein coding regions. Proc. Natl Acad. Sci. USA **88**: 5518–22.

200 Stenson, P. D., E. V. Ball, M. Mort, *et al*. 2003. Human Gene Mutation Database (HGMD): 2003 update. Hum. Mutat. **21**: 577–81.

201 Strausberg, R. L., K. H. Buetow, M. R. Emmert-Buck and R. D. Klausner. 2000. The cancer genome anatomy project: building an annotated gene index. Trends Genet. **16**: 103–6.

202 Suckow, J., P. Markiewicz, L. G. Kleina, J. Miller, B. Kisters-Woike and B. Muller-Hill. 1996. Genetic studies of the Lac repressor. XV: 4000 single amino acid substitutions and analysis of the resulting phenotypes on the basis of the protein structure. J. Mol. Biol. **261**: 509–23.

203 Summers, N. L., W. D. Carlson and M. Karplus. 1987. Analysis of sidechain orientations in homologous proteins. J. Mol. Biol. **196**: 175–98.

204 Summers, N. L. and M. Karplus. 1989. Construction of side-chains in homology modelling: Application to the C-terminal lobe of rhizopuspepsin. J. Mol. Biol. **210**: 785–811.

205 Summers, N. L. and M. Karplus. 1990. Modeling of globular proteins: A distance-based search procedure for the construction of insertion/deletion regions and Pro ↔ non-Pro mutations. J. Mol. Biol. **216**: 991–1016.

206 Sun, M., Z. Li, Y. Zhang, Q. Zheng and C. C. Sun. 2005. Homology modeling and docking study of cyclin-dependent kinase (CDK) 10. Bioorg. Med. Chem. Lett. **15**: 2851–6.

207 Sutcliffe, M. J., I. Haneef, D. Carney and T. L. Blundell. 1987. Knowledge based modeling of homologous proteins, part I: three-dimensional frameworks derived from the simultaneous superposition of multiple structures. Protein Eng. **5**: 377–84.

208 Sutcliffe, M. J., F. R. Hayes and T. L. Blundell. 1987. Knowledge based modeling of homologous proteins, Part II: rules for the conformations of substituted sidechains. Protein Eng. **1**: 385–92.

209 Swindells, M. B., M. W. MacArthur and J. M. Thornton. 1995. Intrinsic ϕ,ψ propensities of amino acids, derived from the coil regions of known structures. Nat. Struct. Biol. **2**: 596–603.

210 Terwilliger, T. C., M. S. Park, G. S. Waldo, *et al*. 2003. The TB structural genomics consortium: a resource for *Mycobacterium tuberculosis* biology. Tuberculosis (Edinb.) **83**: 223–49.

211 Thompson, J. D., D. G. Higgins and T. J. Gibson. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. **22**: 4673–80.

212 Tuffery, P., C. Etchebest, S. Hazout and R. Lavery. 1991. A new approach to the rapid determination of protein side chain conformations. J. Biomol. Struct. Dyn. **8**: 1267–89.

213 Vasquez, M. 1995. An evaluation of discrete and continuum search techniques for conformational analysis of side-chains in proteins. Biopolymers **36**: 53–70.

214 Venclovas, C., A. Zemla, K. Fidelis and J. Moult. 1997. Criteria for evaluating protein structures derived from comparative modeling. Proteins **Suppl. 1**: 7–13.

215 Venclovas, C., A. Zemla, K. Fidelis and J. Moult. 1999. Some measures of comparative performance in the three CASPs. Proteins **Suppl. 3**: 231–7.

216 Vijayasri, S. and S. Agrawal. 2005. Domain-based homology modeling and mapping of the conformational epitopes of envelope glycoprotein of West Nile virus. J. Mol. Model. **11**: 248–55 [Online].

217 Vlijmen, H. W. T. v. and M. Karplus. 1997. PDB-based protein loop prediction: Parameters for selection and methods for optimization. J. Mol. Biol. **267**: 975–1001.

218 Voigt, C. A., D. B. Gordon and S. L. Mayo. 2000. Trading accuracy for speed: A quantitative comparison of search algorithms in protein sequence design. J. Mol. Biol. **299**: 789–803.

219 Vriend, G. 1990. WHAT IF: a molecular modeling and drug design program. J. Mol. Graph. **8**: 52–56.

220 WALLNER, B. AND A. ELOFSSON. 2005. All are not equal: a benchmark of different homology modeling programs. Protein Sci. **14**: 1315–27.

221 WANG, G., J. M. SAUDER AND R. L. DUNBRACK JR. 2005. Comparative modeling in structural genomics. In SUNDSTROM, M., M. NORIN AND A. EDWARDS (eds.), *Structural Proteomics and High Throughput Structural Biology*. CRC Press, New York, NY: 109–36.

222 WATSON, J. D., R. A. LASKOWSKI AND J. M. THORNTON. 2005. Predicting protein function from sequence and structural data. Curr Opin Struct Biol **15**: 275–84.

223 WEBER, I. T. 1990. Evaluation of homology modeling of HIV protease. Proteins **7**: 172–84.

224 WEBER, I. T., M. MILLER, M. JASKOLSKI, J. LEIS, A. M. SKALKA AND A. WLODAWER. 1989. Molecular modeling of the HIV-1 protease and its substrate binding site. Science **243**: 928–31.

225 WHISSTOCK, J. C. AND A. M. LESK. 2003. Prediction of protein function from protein sequence and structure. Q. Rev. Biophys. **36**: 307–40.

226 WILMOT, C. M. AND J. M. THORNTON. 1988. Analysis and prediction of the different types of beta-turn in proteins. J. Mol. Biol. **203**: 221–32.

227 WILMOT, C. M. AND J. M. THORNTON. 1990. Beta-turns and their distortions: a proposed new nomenclature. Protein Eng. **3**: 479–93.

228 WILSON, C., L. GREGORET AND D. AGARD. 1993. Modeling sidechain conformation for homologous proteins using an energy-based rotamer search. J. Mol. Biol. **229**: 996–1006.

229 WINTJENS, R. T., M. J. ROOMAN AND S. J. WODAK. 1996. Automatic classification and analysis of alpha alpha-turn motifs in proteins. J. Mol. Biol. **255**: 235–53.

230 WOHLFAHRT, G., V. HANGOC AND D. SCHOMBURG. 2002. Positioning of anchor groups in protein loop prediction: the importance of solvent accessibility and secondary structure elements. Proteins **47**: 370–8.

231 WOJCIK, J., J. P. MORNON AND J. CHOMILIER. 1999. New efficient statistical sequence-dependent structure prediction of short to medium-sized protein loops based on an exhaustive loop classification. J. Mol. Biol. **289**: 1469–90.

232 WOLF, Y. I., S. E. BRENNER, P. A. BASH AND E. V. KOONIN. 1999. Distribution of protein folds in the three superkingdoms of life. Genome Res. **9**: 17–26.

233 WOLF, Y. I., N. V. GRISHIN AND E. V. KOONIN. 2000. Estimating the number of protein folds and families from complete genome data. J. Mol. Biol. **299**: 897–905.

234 XIANG, Z. AND B. HONIG. 2001. Extending the accuracy limits of prediction for side-chain conformations. J. Mol. Biol. **311**: 421–30.

235 XIANG, Z., C. S. SOTO AND B. HONIG. 2002. Evaluating conformational free energies: the colony energy and its application to the problem of protein loop prediction. Proc. Natl. Acad. Sci. USA **99**: 7432–7.

236 XU, J. 2005. Rapid protein side-chain packing via tree decomposition. Proc. RECOMB **9**: 423.

237 YAMAGUCHI, A., M. IWADATE, E. SUZUKI, K. YURA, S. KAWAKITA, H. UMEYAMA AND M. GO. 2003. Enlarged FAMSBASE: protein 3D structure models of genome sequences for 41 species. Nucleic Acids Res. **31**: 463–8.

238 YANG, J., L. F. TEN EYCK, N. H. XUONG AND S. S. TAYLOR. 2004. Crystal structure of a cAMP-dependent protein kinase mutant at 1.26A: new insights into the catalytic mechanism. J. Mol. Biol. **336**: 473–87.

239 ZEMLA, A., C. VENCLOVAS, K. FIDELIS AND B. ROST. 1999. A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. Proteins **34**: 220–3.

240 ZEMLA, A., C. VENCLOVAS, J. MOULT AND K. FIDELIS. 1999. Processing and analysis of CASP3 protein structure predictions. Proteins **Suppl. 3**: 22–9.

241 ZHANG, C. AND S. H. KIM. 2003. Overview of structural genomics: from structure to function. Curr. Opin. Chem. Biol. **7**: 28–32.

242 ZHANG, P., J. XIE, G. YI, C. ZHANG AND R. ZHOU. 2005. *De novo* RNA synthesis and homology modeling of the classical swine fever virus RNA polymerase. Virus Res. **112**: 9–23.

243 ZHANG, Y., Y. Y. SHAM, R. RAJAMANI, J. GAO AND P. S. PORTOGHESE. 2005. Homology modeling and molecular dynamics simulations of the mu opioid receptor in a membrane-aqueous system. ChemBiochem. **6**: 853–9.

244 ZHENG, Q. AND D. J. KYLE. 1996. Accuracy and reliability of the scaling-relaxation method for loop closure: an evaluation based on extensive and multiple copy conformational samplings. Proteins **24**: 209–17.

245 ZHENG, Q., R. ROSENFELD, C. DELISI AND D. J. KYLE. 1994. Multiple copy sampling in protein loop modeling: computational efficiency and sensitivity to dihedral angle perturbations. Protein Sci. **3**: 493–506.

246 ZHENG, Q., R. ROSENFELD, S. VAJDA AND C. DELISI. 1993. Determining protein loop conformation using scaling-relaxation techniques. Protein Sci. **2**: 1242–8.

247 ZOLLER, B. AND B. DAHLBACK. 1994. Linkage between inherited resistance to activated protein C and factor V gene mutation in venous thrombosis. Lancet **343**: 1536–8.

11
Protein Fold Recognition Based on Distant Homologs

Ingolf Sommer

1 Introduction

In the 1960s Anfinsen showed with a rather simple experiment that for many proteins the sequence is the sole determinant for the three-dimensional (3-D) structure [7, 8]. A denaturating substance was added to the solution of a protein, resulting in the loss of native protein structure and function. After removal of the denaturing substance, proteins recovered the functional activity (an enzymatic reaction in Anfinsen's experiment). Thereby, it was concluded that the protein managed to refold itself in the absence of any other agents.

Later, it became evident that some proteins need other proteins to fold and some proteins or parts of proteins remain unfolded (see Chapter 9). Still, the Anfinsen principle has been a guiding force for much research that aims at understanding mechanisms for determining the 3-D structure of proteins given their sequence. Although great improvements of these methods have been achieved, the protein structure prediction problem is still unsolved, in general. The folding process that determines the structure is not known to enough detail to serve as a basis for modeling. Instead, prediction methods have to rely on heuristic inductive inferences.

One very successful approach to the prediction of protein structures today models the protein structure based on another structurally resolved protein as a structural template. We call this approach template-based modeling. The protein whose structure is to be predicted is called the target. In addition to the target sequence, the method requires the input of a database of resolved protein structures, the so-called template structures. Rather than modeling the protein structure *de novo*, we repeatedly ask the question whether the protein structure of the target sequence is similar to a template structure. This leaves us with a set of candidates for templates after which we can model the structure of the target sequence.

This chapter deals with the question how likely it is that a target protein sequence attains a structure similar to a given template structure. In principle

Bioinformatics – From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

one can compare the sequence itself to the sequences of the template structures (sequence–sequence comparison, Section 3.1), additionally take evolutionary information into account (profile methods, Sections 3.2 and 3.3) or thread the sequence onto the given 3-D template structure taking physico-chemical properties of the template structure into account (Section 4). The problem of identifying suitable templates typically becomes harder the more distant the target sequence is related to its most similar sequences in the template database. More similar sequences are easier to identify when looking at sequence information only; typically, they also have a more similar structure within a chemically more similar environment.

Traditionally, protein structure prediction has been divided into homology modeling (also called comparative modeling; see Chapter 10), fold recognition (this chapter) and prediction of novel folds (Chapter 12) [97]. In homology modeling, closely homologous templates are available affording very precise models for the protein structure. In fold recognition, identifying a suitable template becomes a challenge. Once a template is obtained, a prediction of the 3-D arrangement can be made, whereas a constructed full-atom model is not reliable, in general. In contrast, in the new fold category no suitable template is available and fragment assembly or *de novo* methods need to be applied.

The basic pipeline exercised is identical in the first two categories, in principle. Thus, today homology modeling and fold recognition are merging. The focus in homology modeling is more on obtaining detailed models with high resolution. The focus of fold recognition is more on the identification of suitable templates.

2 Overview of Template-based Modeling

2.1 Key Steps in Template-based Modeling

The input to template-based modeling is a target sequence and a database of previously resolved template structures. The output is a 3-D model for the target sequence, constructed according to the template.

2.1.1 Identifying Templates

How do we decide how likely the sequence of a target protein attains a given template structure? We perform a pairwise alignment of the target sequence with the sequences of the template structures. Here, different techniques can be used: the target sequence can be aligned with the amino acid sequences of the templates. If evolutionary information on the target sequence is available (see Chapters 3 and 4), a multiple alignment of related sequences can be used to construct a position-specific scoring matrix (PSSM or, similarly, a frequency profile, see Section 3.2.1). A PSSM represents the preferences of a residue in the target sequence to be matched with a residue in the template structure (Section 3.2.1). This kind of evolutionary information can also be used on the template side or on both target and template sides. One can enhance this approach by introducing additional information, e.g. stemming from secondary structure predictions (Section 5.1.1). Sequence–structure alignment methods using the 3-D structure information on the template sides during the template identification process are presented in Section 4.

2.1.2 **Assessing Significance**

The score of the alignment tells us the propensity of the target sequence to attain the template structure. Ideally, the template that can be aligned to the target sequence with the highest alignment score should provide the structural model for the target protein. However, since alignment scores reflect structural preference only inadequately, this model selection procedure is fallible. Thus, the score has to be accompanied with a confidence value that rates how much we can trust the prediction. Often, a confidence score is based on theoretical statistical significance which rates how unlikely it is to obtain the alignment by chance. In addition, empirical choices of confidence values have also proven effective (Section 6.1).

2.1.3 **Model Building**

Aligning the target sequence to a template protein is only the first step of producing a full-atom protein structure model for the target protein. While it is the critical step for the similarity range of below about 40% sequence identity between target and template sequence, it is less difficult in high similarity ranges (see Chapter 10). An incorrect alignment invariably leads to a wrong protein model.

The alignment maps residues of the target onto the template. An initial model is constructed by copying coordinates of the template structure and changing the template residue types according to the target sequence. This model only provides a part of the structure of the target protein's backbone. Gaps in the alignment represent parts of the target sequence that we cannot map onto the template structure (if gaps occur in the template sequence) or tears and rips in the backbone model of the target sequence (if gaps occur in the target sequence). The former gaps mostly coincide with loops in the target protein that have no counterpart in the template structure. These loops have to be modeled in a separate loop modeling step by inserting loop fragments from a database of protein structures, or by using energy optimization methods. Rips in the backbone of the target protein have to be mended and, finally, the sidechains of the target protein have to be attached to the backbone. Here, the variants are to use a database of side chain rotamers or to do energy optimization. Similarity-based protein modeling tools combine these steps in different ways and using different algorithmic procedures (see Chapter 10) [32, 129, 133].

2.1.4 Evaluation

The performance of fold recognition methods is typically assessed by benchmarks [20, 37, 71, 171]. Here, the objective is to retrieve the template structure that is most similar to the structure of the target protein. The performance can be quantified in terms of the number of correctly assigned folds or, in a more detailed fashion, by rating the quality of the alignment, on which the fold assignment is based. The accuracy of protein sequence–structure alignment methods depends highly on the sequence similarity between the target and the template protein. It is low in the case of low sequence similarity and high in the case of high sequence similarity. Over the whole protein structure database we can today achieve an accuracy of above 70% correctly assigned folds [165]. Starting 1994, the Critical Assessment of Protein Structure Prediction (CASP) experiment, a biannual blind test for protein structure prediction methods was established to measure progress in the field [98].

While similarity-based modeling is quite successful, this approach cannot discover yet unseen protein structures. Rather, it can only rediscover structures that have been seen before as attained by different protein sequences. As we can only assume to have uncovered about one half of all protein folds used by nature [72, 178], this approach has strong limitations.

2.2 Template Databases

Typically the structures serving as templates are experimentally determined by X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy.

X-ray crystallography determines structures of macromolecules by analyzing their diffraction patterns when irradiated by X-rays. In order to obtain a diffraction pattern a protein has to be crystallized. During crystallization many instances of the same protein are symmetrically arranged along a lattice. Even when crystallized, proteins frequently display biological activity indicating that the crystallization process captured them in the biologically active form. Some proteins are very hard to crystallize (e.g. membrane proteins). When the protein crystal is properly irradiated with X-rays, diffraction patterns result that are captured on film or digital media. From these patterns the electron density of the protein is computed and from the density information the atomic structure of the protein is derived [13, 14, 120].

NMR spectroscopy is the second prominent method for structure determination. It can be applied to smaller proteins (up to around 40 kDa) in highly concentrated solution. Some atomic nuclei, e.g. protons in hydrogen, display an intrinsic magnetic property called spin. By applying an external varying magnetic field the state of the spin may change, provided resonance takes place. A resonance spectrum can be obtained. The surrounding local environment of nuclei may cause a shift of the peaks in the resonance spectrum. Peaks in the spectrum are associated with residues in the protein sequence in a sequential assignment step and a list of distance constraints can be deduced. Applying distance geometry techniques to these data, atom coordinates can be estimated [13, 14, 162].

The most frequently used public resource for coordinates of protein structures determined with NMR spectroscopy or X-ray crystallography is the Protein Data Bank (PDB). Quality of structures deposited in the PDB can be judged with a number of programs (Procheck [94], Whatcheck [55]). The ASTRAL compendium for structure and sequence analysis conveniently combines the output of several of these programs with additional manual annotation into a joint AEROSPACI score [16, 24].

Most proteins are globular. Larger proteins often fold into several independent folding units or domains. Domains are compact regions of structure often capable of folding on their own in aqueous solution. Domains can be defined as folding units, as units of structural similarity, or as evolutionary units [115]. Several resources describe the domain composition within proteins and classify the identified domains hierarchically: most prominent are SCOP, CATH and FSSP. SCOP, (structural classification of proteins [99]) is a human-curated database organized hierarchically into classes, folds, superfamilies and families. CATH (class, architecture, topology and homologous superfamily [106]) is a semi-automated procedure. The FSSP (families of structurally similar proteins [54]) relies on a fully automatic procedure.

Since domains are recurring folding units, often domains are chosen as templates for structure prediction. For the construction of template databases, reasonably different templates with high quality structures are favored. Representative sets of templates can be constructed limiting the maximal percentage of sequence identity while choosing structures with a high SPACI score for example. Choosing SCOP domains with at most 40% (95%) sequence identity results in a set of 7290 (12 065) structures for SCOP version 1.69 of July 2005.

3 Sequence-based Methods for Identifying Templates

3.1 Sequence–Sequence Comparison Methods

The simplest method for assigning a fold to a target sequence is to compare the target sequence to sequences of proteins with known structures. In order to compare the sequences they are aligned (see Chapter 3, [34]).

Scoring the similarity of an individual pair of amino acids amounts to comparing the likelihoods of generating this pair from two alternative stochastic models. One model, the model of related amino acids, describes the distribution of amino acid pairs originating from related positions in pairwise alignments of homologous sequences. The second model is a null model which describes the distribution of unrelated amino acid pairs. The amino acid pair is denoted by a pair of random variables (X, Y) both with values in $\{1, \ldots, 20\}$. The distribution of (X, Y) under the related model is defined as $p_{\mathrm{rel}}(i, j)$. Note that the background distribution of amino acids can be computed as $p_i = \sum_{j=1}^{20} p_{\mathrm{rel}}(i, j)$. The log-likelihood ratios of all pairs of amino acids are stored in an amino acid similarity matrix: $M_{i,j} = \lambda \log \left(\frac{p_{\mathrm{rel}}(i,j)}{p_i p_j} \right)$, with constant scaling factors λ. Different factors were introduced by different authors: Dayhoff and coworkers [29] use $\lambda = 10/\log 10$, the BLOSUM series [51] uses $\lambda = 2/\log 2$.

Introducing an additional penalty for inserting and extending gaps, two sequences of unequal length can be aligned. For arbitrary sequences, the Needleman–Wunsch algorithm [100], aligns two sequences using dynamic programming. It is appropriate, when sequences are expected to be similar from beginning to end, computing a so-called *global* alignment. In cases where only limited patches of the sequences share similarity, *local* alignment is used, which is computed using the Smith–Waterman algorithm [145]. When searching for similarity of a complete subsequence within a sequence, as is the case for example when identifying domains within longer sequences, *free-shift* alignments are appropriate (domain identification is also discussed in the context of Chapter 12).

The runtime of alignment algorithms is measured in terms of (sum of the) length of the aligned sequences. Using the Needleman–Wunsch algorithm, alignments can be computed in cubic time for general gap cost functions. For affine gap-cost functions, alignments can be computed in quadratic time using the Gotoh algorithm [48]. Since these dynamic programming programming algorithm can be slow for high-throughput applications fast heuristics like BLAST [5] and FASTA [113] were developed which approximate the optimal alignment algorithms.

The parameters of sequence alignment methods have to be calibrated for the intended application, e.g. Pearson details a protocol for searching genomes with these methods [111]. The major disadvantage of these pairwise sequence comparison methods is that conserved and variable positions are treated indifferently and contribute equally to the final alignment score. This limits the ability to identify distant homologs.

Sequence–sequence comparison methods like the Smith–Waterman algorithm [145] and the BLAST [5] or FASTA [113] tools can assign a fold to approximately 20–30% of the proteins coded by genes in a microbial genome [45, 173].

3.2 Frequency Profile Methods

Exploiting evolutionary information in addition to the plain sequence information, frequency profile methods are powerful tools for detecting distant relationships between amino acid sequences, often picking up signals even when other methods fail [170]. In this section we define frequency profiles, and describe why one would use them, how they are generated and different ways of comparing profiles to sequences and profiles to other profiles.

3.2.1 Definition of a Frequency Profile and PSSM

A frequency profile (or profile for short) is a sequence of position-specific frequencies of amino acids, which can be used for sequence alignments instead of a sequence containing individual amino acids at each position [49, 150].

Starting with an alignment of several related sequences (a multiple sequence alignment), for each position in that multiple alignment the occurrences of the types of amino acids are counted (for an example, see Figures 1 and 2). This yields an estimate of the likelihood of different types of amino acids occurring at different positions along the sequence. By counting gap characters in an alignment column, information on the likelihood of insertions at individual positions along the sequence can be gathered. Before counting, each sequence can be given a weight, which is a useful debiasing procedure if several of the sequences are very similar [49, 52, 91, 134].

Formally, a frequency profile matrix $(F_{p,a})$ is composed of at least 20 columns and N rows. The row p corresponds to the sequence position in the multiple alignment of length N and the column a to the type of amino acid. The first 20 columns of each row specify the relative frequencies of the types of amino acids at that position [123, 167, 185]. Some scoring schemes require additional columns containing penalties for insertions or deletions at that position [49], or unexpected characters in the sequences [18].

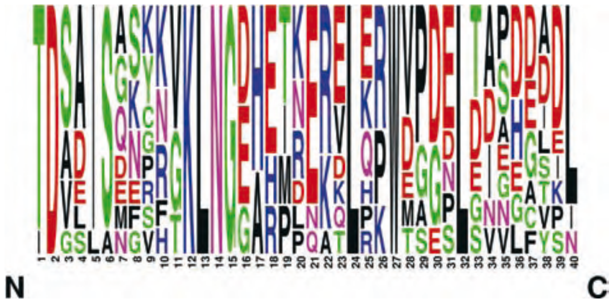


Figure 1 Sequence logo corresponding to the first 40 sequence positions of Pfam [11] multiple alignment of the Ataxin-2 N-terminal region. The figure is produced with WebLogo [28]. The overall height of the stack at each sequence position indicates the sequence conservation at that position, while the height of symbols within the stack indicates the relative frequency of each amino acid at that position.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A			0.2	0.5		0.1	0.2										0.3			
C									0.1											
D		1.0		0.2			0.1									0.4				0.1
E				0.1			0.1	0.1								0.4		0.6		
F								0.1		0.1										
G			0.1				0.2	0.1	0.1		0.3				1.0	0.2				
H										0.1							0.7	0.2		
I	0.1				0.9														0.2	
K								0.2	0.2	0.3		1.0								0.3
L				0.1	0.1								1.0							0.1
M							0.1												0.2	
N							0.1	0.2		0.2				1.0						0.2
P									0.1										0.2	0.1
Q							0.2													
R									0.1	0.2								0.2		0.2
S			0.5	0.1		0.9		0.3	0.1											
T	0.9										0.2								0.4	
V			0.2						0.1		0.5									
W																				
Y									0.2											

Figure 2 First 20 sequence positions of the frequency profile matrix corresponding to the sequence logo depicted in Figure 1. All sequences are weighted identically.

Originally, Gribskov [49] combined the frequency profile matrix directly with amino acid substitution matrices [29,51], yielding

$$(S_{p,a}) = \left(\sum_{b=1}^{20} F_{p,b} \cdot M_{a,b} \right),$$

where $(M_{a,b})$ is Dayhoff's substitution matrix. This construct is referred to as a PSSM. In contrast to a frequency profile, a PSSM is a frequency profile multiplied with substitution preferences.

Other authors also use the word *profile* in the context of structure-based template identification methods to describe the sequence-based information extracted from protein structures, as discussed in Sections 4 and 4.3, in particular.

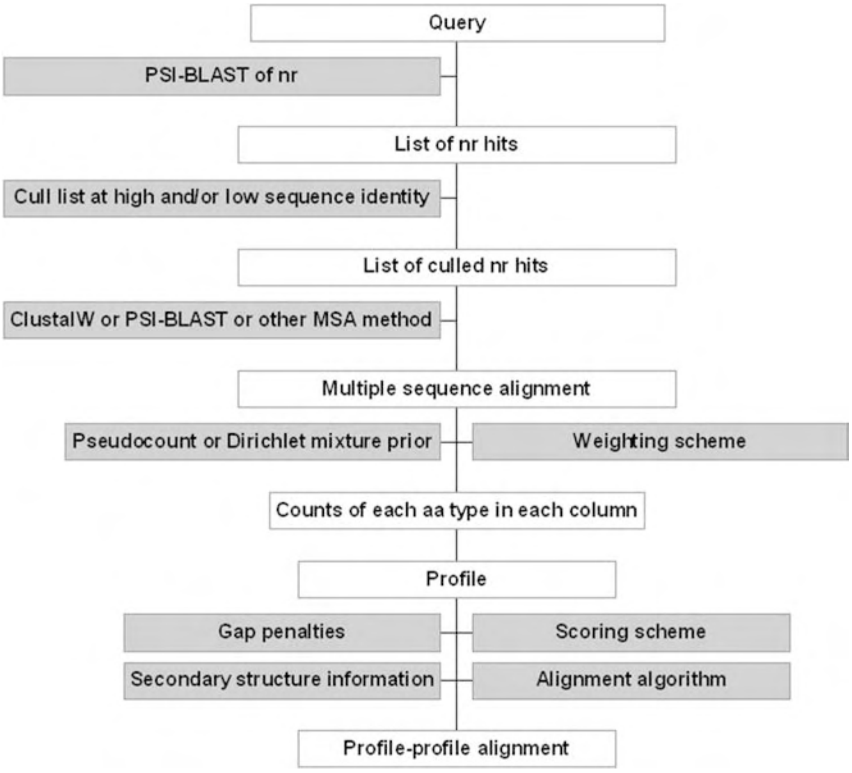

Query
PSI-BLAST of nr
List of nr hits
Cull list at high and/or low sequence identity
List of culled nr hits
ClustalW or PSI-BLAST or other MSA method
Multiple sequence alignment
Pseudocount or Dirichlet mixture prior
Weighting scheme
Counts of each aa type in each column
Profile
Gap penalties
Scoring scheme
Secondary structure information
Alignment algorithm
Profile-profile alignment

Figure 3 Illustration of the generation and application of profile alignments (from Wang and Dunbrack [171], reprinted with permission of Cold Spring Harbor Laboratory Press and the authors).

3.2.2 Generating Frequency Profiles

For constructing a profile one needs a multiple alignment of related sequences. Such alignment can be readily available, e.g. as in Pfam [11]. If not, one can start generating a profile from a single sequence by searching related sequences and multiply aligning them. Once a few sequences are found, a profile can be constructed from them and employed to search more sequences. This iterative approach for searching homologs is implemented in the popular PSI-BLAST program [6, 131], for example. In practice, the number of sequences identified by this search matters. This number is controlled by the BLAST parameters "number of iterations" and "E-value" for inclusion into the PSSM. The multiple alignment can be taken directly from the BLAST output, iteratively optimized [136], or improved with tools for multiple sequence alignment like ClustalW [156] or T-Coffee [101, 107]. The process of profile generation is illustrated in Figure 3.

If one particular sequence is the seed for searching further sequences, the multiple sequence alignment can be cropped by deleting the columns which contain gaps in the master sequence (master–slave alignment in BLAST terminology). Such an alignment can be used to construct a profile specifically reflecting that sequence. In the matrix notation above, this results in a frequency profile matrix F with rows eliminated until the N remaining rows correspond to the N residues of the master sequence.

Profiles calculated from multiple alignments that originate from similarity searches are subject to a bias introduced by the composition of the sequence databases that are searched [3]. An example is a case such that a search yields a large number of hits of mammalian origin and only few distantly related plant sequences. If a frequency profile from such a multiple alignment were calculated by simply taking the relative frequencies of the amino acids, this would reflect a strong emphasis on the mammalian sequences. This effect is not desirable, since the goal is to present all sequences of the family in an unbiased manner in the profile. To tackle this problem, it is generally assumed that only a fraction of the sequences of the family to be modeled is available in the databases and many sequences have not been observed, so far. A number of methods has been developed to estimate the size of the sequence families from the available sample. The most important are Dirichlet mixture models [17], pseudocounts [153], minimal-risk estimation [180] and sequence weighting models [52,74,89,123,125,134,150]. For a discussion of sequence weighting models, see Ref. [171].

3.2.3 Scoring Frequency Profiles

Frequency profiles can be aligned using the same algorithms as in sequence alignment. Whereas in plain sequence alignments two individual amino acids are matched, here profile vectors are compared. Profile vectors can be matched to individual amino acids or to other profile vectors. Different schoring schemes exist for both approaches .

3.2.4 Scoring Profiles Against Sequences

Let α be a row-vector at a certain sequence position p in a profile F and let $a_i = F(p,i)$ be the frequency observed for amino acid number i at position p. In the following, we will investigate ways of scoring α against an individual amino acid of type j (profile–sequence), or against another row-vector β from another profile F' (profile–profile).

The average score was the first profile–sequence score used in bioinformatics [22,49]. It is defined as

$$\text{score}_{\text{avg}}(\alpha, j) = \sum_{i=1}^{20} \alpha_i M_{i,j}.$$

Its basic idea is to compute the expected value of the sequence–sequence score under the profile distribution. Later, we will see an extension of this idea to profile–profile scoring. This scoring system has several drawbacks which led to further development.

The previously mentioned iterative PSI-BLAST program [6] starts with a round of BLAST using sequence–sequence alignment techniques to identify related sequences. From the sequences found, a profile is constructed which is used to identify further sequences using essentially the average scoring scheme. No position-specific gap costs are used, instead for each iteration the same gap costs that are used in the initial BLAST run are applied. In contrast to PSI-BLAST, the related tool IMPALA [131] implements the opposite sequence–profile direction and compares a single target sequence to a database of PSSMs previously generated with PSI-BLAST. This method can be used to quickly compare a sequence to a database of template structures with precomputed PSSMs.

Log-likelihood profile–sequence scoring (e.g. Ref. [27]):

$$\text{score}_{\text{lq}}(\alpha, j) = \log \frac{\alpha_j}{p_j},$$

is an optimal (in the sense that the likelihood ratio guarantees the lowest error of type II of all tests at the same level of significance) test statistic according to the Neyman–Pearson lemma from statistical test theory for deciding whether the amino acid j is a sample from the distribution α or rather from the background distribution p. The values summed up over all aligned positions provide a direct measure of the likelihood of the amino acid sequence being a sample from the profile.

Evolutionary profile–sequence scoring [27,50] is defined as

$$\text{score}_{\text{ev}}(\alpha, j) = \log \sum_{i=1}^{20} \alpha_i \frac{p_{\text{rel}}(i, j)}{p_i p_j}.$$

This score summed up over all aligned positions in an alignment is an optimal means of deciding whether the sequence occurs by chance or is more likely to be the result of sampling from the profile having undergone evolutionary transition.

3.2.5 Scoring Profiles against Profiles

Since profiles have been invented, several ways of comparing profiles to profiles have been developed and tested. These scoring schemes perform differently in their abilities for searching templates and in the quality of the alignments produced.

A simple, fast and heuristic way of comparing two profile vectors independently of any substitution matrix is the dot product scoring, which is the summation of the products of the frequencies per type of amino acid [117,123,124]:

$$\text{score}_{\text{dotprod}}(\alpha, \beta) = \sum_{i=1}^{20} \alpha_i \beta_i .$$

Average or cross-product profile–profile scoring is a straightforward extension of the approach used in profile–sequence scoring. The products of the frequencies are multiplied with the corresponding log–odds elements of the substitution matrix:

$$\text{score}_{\text{avg}}(\alpha, \beta) = \sum_{i=1}^{20} \sum_{j=1}^{20} \alpha_i \beta_j \log \frac{p_{\text{rel}}(i, j)}{p_i p_j} .$$

Though not called profile–profile alignment, this approach has been used in ClustalW [155, 156], where two multiple alignments are aligned using the average over all pairwise scores between residues.

The log-average scoring multiplies the products of the frequencies by the corresponding probabilities in the substitution matrix and then takes the logarithm [165, 167]:

$$\text{score}_{\text{logavg}}(\alpha, \beta) = \log \sum_{i=1}^{20} \sum_{j=1}^{20} \alpha_i \beta_j \frac{p_{\text{rel}}(i, j)}{p_i p_j} .$$

This scoring is symmetric and scores zero against the background distribution. For the special case of profile–sequence alignment, i.e. for one of the profiles corresponding to just one sequence, log-average scoring reduces to score_{ev}. For the special case of both profiles corresponding to one sequence, log-average scoring reduces to the scoring in normal sequence–sequence alignment.

The Jensen–Shannon divergence $D^{\text{JS}}_{\lambda}(\alpha, \beta) = \lambda D^{\text{KL}}(\alpha, \gamma) + (1 - \lambda) D^{\text{KL}}(\beta, \gamma)$, with $\gamma = \lambda\alpha + (1 - \lambda)\beta$ for $0 \leq \lambda \leq 1$ is based on the information theoretic Kullback–Leibler distance $D^{\text{KL}}(\alpha, \beta) = \sum_{k=1}^{20} \alpha_k \log_2 \frac{\alpha_k}{\beta_k}$. With p as amino acid background distribution, Yona and Levitt developed the score [185]:

$$\text{score}_{\text{JensenShannon}}(\alpha, \beta) = \frac{1}{2}(1 - D^{\text{JS}}(\alpha, \beta))(1 + D^{\text{JS}}(\gamma, p)).$$

Given two profile vectors α and β, the corresponding row vectors of the PSI-BLAST PSSMs S and T, and the effective number of observations n and m, the Panchenko score [108] is computed as:

$$\text{score}_{\text{Panchenko}}((\alpha, S, m), (\beta, T, n)) = \frac{m\,\alpha \cdot T + n\,\beta \cdot S}{n + m} .$$

The LogOddsMultin score used in the COMPASS tool for comparison of multiple protein alignments [125] is an extension of the scoring that PSI-BLAST uses:

$$\text{score}_{\text{LogOddsMultin}} = c_1 \sum_{i=1}^{20} n_i^{(1)} \log \frac{q_i^{(2)}}{p_i} + c_2 \sum_{i=1}^{20} n_i^{(2)} \log \frac{q_i^{(1)}}{p_i} ,$$

where $n_a^{(1)}$ and $n_a^{(2)}$ are the effective counts for each amino acid in columns 1 and 2, and where $c_1 = \frac{n^{(2)}-1}{n^{(1)}+n^{(2)}-2}$, $c_2 = \frac{n^{(1)}-1}{n^{(1)}+n^{(2)}-2} = 1 - c_1$ and $n^{(k)} = \sum_{j=1}^{20} n_j^{(k)}$.

These scoring schemes are extensively discussed and experimentally compared in Refs. [164, 171]. Both studies compare and analyze the searching abilities as well as the quality of the alignments produced with these scorings schemes. Alignment quality can be monitored with measures like Q_{Modeler} as fraction of correctly aligned positions in the profile–profile alignment [130], $Q_{\text{Developer}}$ as fraction of correctly aligned positions in the structural alignment [130], or similarly Q_{Combined} that penalizes sequence alignments that are either too long or too short [185]. Measuring alignment quality with Q_{Combined}, the above profile–profile scoring functions perform similarly, while displaying differences in Q_{Modeler} and $Q_{\text{Developer}}$ [171]. The Jensen–Shannon and LogOddsMultin functions produce shorter, more accurate alignments [171]. The Jensen–Shannon scoring produces better alignments for closely related sequences [164]. Of the scoring functions above, in terms of search specificity and sensitivity the LogOddsMultin and log-average perform significantly better [171].

3.3 Hidden Markov Models (HMMs)

3.3.1 Definition

HMMs (see Chapter 3 or Ref. [34]) can be regarded as a generalization of profiles and have become an important tool in fold recognition. Introduced in the late 1960s and 1970s, and popular in speech recognition [118], HMMs made it into computational biology in the late 1980s [26] and have been used as profile models since the mid-1990s [73, 151]. For an introduction to HMMs, see Chapter 3 or Refs. [9, 118]; for review articles in the context of computational biology, see Refs. [25, 35].

HMMs are probabilistic models that are applicable to signals, time series or linear sequences. An HMM is a system characterized by the following: It has a set of hidden states $S = \{S_1, S_2, \ldots, S_N\}$. The system is in state q_t at time t and has a number M of distinct observation symbols per state, i.e. a discrete alphabet $V = \{v_1, v_2, \ldots, v_M\}$. The system randomly evolves according to a state transition probability distribution matrix $A = (a_{ij})$, where $a_{ij} = P(q_{t+1} = S_j | q_t = S_i)$ for $1 \leq i, j \leq N$ and emits characters from the alphabet V according to an emission probability matrix $E = (e_{ik})$, $1 \leq i \leq N, 1 \leq k \leq M$. When the system is in a given state i, it has a probability a_{ij} of moving to state j and a probability e_{ik} of emitting letter v_k. Biological relevance is attached to the hidden states of the Markov model. Transition and emission probabilities depend on the current state only and not on the past. This property is called the first-order Markov assumption. Only the emitted symbols can be observed, not the underlying random walk from state to state.

3.3.2 Profile HMM Technology

When applying HMMs to model families of protein sequences one speaks of profile HMMs. In this case, mainly an alphabet of 20 amino acid letters is used, but also other alphabets exist such as 64-letter alphabets for codon triplets, three-letter alphabets (helix, sheet and coil) for secondary structure prediction and Cartesian products of these alphabets. If necessary, meta-characters such as gap symbols can be added to the alphabets, as well.

In a standard architecture for protein sequence HMMs there are three classes of states [9], besides the *start* and *end* state: the *match*, *insert* and *delete* states. Thus $S = \{start, m_1, \ldots, m_N, i_1, \ldots, i_{N+1}, d_1, \ldots, d_N, end\}$. Typically, the length of the model N is the average length of the sequences in the family. The *match* and *insert* states always emit amino acid symbols, whereas the delete states never do. The basic path of state transitions is *start* $\rightarrow m_1 \rightarrow m_2 \rightarrow \ldots \rightarrow m_N \rightarrow$ *end*. Each *match* state m_j has an outgoing edge to succeeding *match*, *delete*, and *insert* states, m_{j+1}, d_{j+1} and i_{j+1}, respectively. Each *delete* state d_j has outgoing edges to the succeeding *delete*, *match* and *insert* states d_{j+1}, m_{j+1} and i_{j+1}. The *insert* state i_j is connected to the succeeding *match* and *delete* states m_{j+1} and d_{j+1}, and has a loop to itself allowing for multiple insertions.

There are three typical questions associated with HMMs in this context [118]. Given an observation sequence of emission characters and a model, how likely is that sequence for the particular model? Given the observation, i.e. target, sequence and a model, what is the underlying sequence of states? How to adjust the model parameters so that the model best describes a multiple alignment of amino acid sequences?

With profile HMMs [73], sequence families can be characterized. For a profile HMM, the first question addresses the likelihood of a given amino acid sequence to be a member of the given family.

For computing the likelihood of a sequence being emitted by a model, the forward procedure [9, 118] is used. Probabilities of states are propagated through the model from the *start* to the *end* states. From the state probabilities, the probabilities of a certain letter being emitted at a certain time can then be computed.

The most likely path of states is computed with the Viterbi algorithm [9, 118]. The path consists of a sequence of matches, insertions and deletions, and thus corresponds to an alignment of the target sequence with the sequence family.

Commonly, with a given multiple alignment the parameters of the standard architecture are initialized prior to learning as follows: The *match* state is assigned to any column of the alignment that contains less than 50% gaps. *Delete* states are associated with columns that contain any gaps. Columns with more than 50% gaps are assigned to corresponding *insert* states. Emissions of *match* and *insert* states can be initialized from the frequency counts of the corresponding columns in the multiple alignment. These counts need to be regularized with Dirichlet distributions or Dirichlet mixtures in order to avoid emissions associated with zero counts [9]. In a subsequent *learning* phase, the HMM parameters are optimized. Various algorithms are available for that, including the expectation maximization (EM or Baum–Welch) algorithm and different generalizations of it as well as gradient descent methods [9, 10, 30, 118]. This learning process corresponds to the generation of a profile in the previous section on profile–profile scoring.

3.3.3 HMMs in Fold Recognition

HMMs have been used for a number of years in fold recognition [25, 35, 66–68] and have been extensively tested (e.g Ref. [110]). One approach [66] is to iteratively add homologous sequences to a HMM (like the PSI-BLAST approach does for sequence alignments). Similar to profile methods, HMMs can turn multiple sequence alignments into position-specific scoring systems suitable for searching databases for remotely homologous sequences [35].

3.3.4 HMM–HMM Comparisons

Similar to the scoring schemes described in the section on frequency profile scoring, there are several ways to score profile HMMs against sequences or against other profile HMMs. Whereas the standard HMM approach was to compare an HMM to one sequence, Lyngsø and coworkers developed an algorithm for the alignment of two HMMs based on the maximization of the coemission probabilities [90]. Recently, Edgar and Sjölander proposed an approach to align two multiple alignments by constructing a profile HMM from one of the alignments and aligning the other to that HMM [36]. Söding has generalized the log-likelihood score maximized in HMM sequence alignments to the case of HMM–HMM alignments [146].

3.4 Support Vector Machines (SVMs)

3.4.1 Definition

SVMs are a state-of-the-art machine learning method for classification problems. SVMs have been succesfully applied for fold recognition.

Conceptually, SVMs map the data points to be classified into a high-dimensional space called feature space, in which an optimally separating hyperplane is sought that separates the two classes of points to be distinguished by the (binary) classification [132]. Technically, the transformation into the high-dimensional space can be avoided and only inner products in that space, called kernels, need to be computed. While the SVM machinery is fairly standardized, kernel functions are highly problem specific. One problem for protein sequences is to map the sequences of typically differing lengths into a space with a constant dimensionality. Several kernel functions exist for the protein classification problem.

3.4.2 **Various Kernels**

Jaakkola and coworkers suggest the Fisher kernel function, which is specific to a protein family [57]. An HMM is trained from positive samples of the family. The so-called Fisher score is the gradient of the log likelihood score for an arbitrary sequence X with respect to the HMM parameters. This score maps the sequence X into a fixed length vector. The Fisher kernel function is then computed on the basis of Euclidean distances between the Fisher score vector for X and the score vectors for known positive and negative examples of the protein family.

Another example, the SVM-pairwise kernel, uses the Smith–Waterman algorithm to align a new protein sequence X against all n sequences in the training set. The feature vector corresponding to protein X is $F_X = f_{x1}, f_{x2}, \ldots, f_{xn}$, where f_{xi} is the E-value of the Smith–Waterman score between sequence X and the ith training set sequence [81].

There are numerous other kernels to treat protein sequences, e.g. mismatch kernels [79], string kernels [126] or motif kernels [12].

3.4.3 **Experimental Assessment**

Machine learning methods need sufficient training data, which imposes some restrictions on the experiments. The typical problem tackled by the machine learning community is the protein classification problem, as described by Jaakkola and coworkers in 1999 [56]. The protein classification problem is to predict the SCOP structural class of a protein given its amino acid sequence. Two sequences with the same superfamily are considered as related by homology and two domains from different folds are considered as unrelated. Proteins from different superfamilies within the same fold have an uncertain relationship, and are not considered in experiments. The classification question then is to decide whether a protein belongs to a certain superfamily.

In the standardized experimental setup defined by Jaakkola and coworkers, and typically used, for each family the protein domains outside the family, but within the same superfamily are taken as positive training samples. Positive test samples are the members of the family. Negative samples are taken from outside the fold to which the family belongs. The set of negative samples is randomly split into training and test samples.

Jaakkola suggests to use only families with at least five family members (positive test) and 10 superfamily members outside the family (positive train). Liao and coworkers [81] suggest using only families with at least 10 family members and five superfamily members outside the family. Both restrictions imply a drastic reduction of the number of families considered at all.

A distincion is made whether all the training data are labeled (supervised learning) or not (semi-supervised learning). The additional data in semi-supervised learning can help to better structure the space around the labeled points. Typically, semi-supervised learning is more expensive and experimentally performs better than supervised learning [177].

Despite the promising developments, it has to be clearly pointed out that SVMs operating directly on the amino acid sequence with kernels like the ones mentioned above currently are not actively used to identify suitable templates. The machine learning methodology requires a certain amount of data for training. Therefore, the protocols used for testing SVMs on the fold recognition problem vary slightly, but significantly, from protocols used to evaluate other current methods. For instance, in CASP6 (see Section 6.3) there was no prediction group relying on kernel methods for template identification.

4 Structure-based Methods for Identifying Templates

Often proteins share similar structure while showing very little (15% or less) sequence identity [62]. Sequence similarity is not necessary for structural similarity, instead extreme divergences of sequences are observed as well as convergent evolution where similar 3-D folds are adopted several times.

Starting in the 1990s, methods were developed for template identification based on structural information such as secondary structure, burial patterns or side-chain pair-interactions. These methods are often referred to as inverse folding, sequence–structure alignment or threading methods. Note, that the term “threading” is widely used to label any method which attempts to tackle the problem of aligning two protein sequences the structure of one of which is known. Originally invented by Jones and coworkers [62] and also frequently used is an alternative definition, according to which “threading” is the alignment of a sequence with a protein structure in 3-D without regard to the sequence associated with the structure [58].

Sequence–structure alignment methods should be able to recognize not only homologous proteins but also analogous proteins sharing the same fold [15, 62]. Most of these methods employ inverse Boltzmann statistics: the frequencies of observed findings are converted into a pseudo-energy function which is optimized in order to find a good sequence–structure alignment (see Section 4.1).

Basically there are two ways of scoring a sequence against a structure. One is to score the interactions of pairs of amino acids of the target sequence within the template structure using pair-interaction potentials as energy functions. Unfortunately, finding a global optimum for this kind of problem has been shown to be NP-hard [77]. There are several approaches to tackle this and we will present four conceptually important ones in Section 4.2.

Alternatively, locations of residues may be evaluated with respect to their placement inside the original template structure afforded by sequence alignments of target and template protein. In this case the global optimum sequence–structure alignment can be found using dynamic programming methods. Standard methods for this are reviewed in Section 4.3.

Today, hybrid algorithms, combining sequence frequency profile methods with sequence–structure methods, are common practice. Either profiles can be integrated into the threading process or methods are exercised separately and the results are merged afterwards. Reviews can be found in Refs. [59, 70, 149, 160]; however, we will first focus on the basic principles.

4.1 Boltzmann's Principle and Knowledge-based Potentials

In protein kinetics, the topography of the landscape of free energies is often described as a funnel [31]. Typically, the energy is assumed to decrease as the folding process proceeds. This organization of the energy landscape is not characteristic of random polypeptides, but is a result of evolution. A common assumption is that the native structure is the one with the lowest free energy. However, many factors contribute to the free energy of the system. Not all factors are known and neither is their interplay completely understood. Therefore, energy landscapes cannot be determined exactly [105].

In addition, in threading we are actually dealing with two structures, i.e. the one to be identified and the one serving as template. The environments of the two structures can be substantially different, making it difficult to apply the same detailed energy potentials to both structures. In spite of these problems, knowledge-based potentials have been employed successfully in threading [19]. Information on different levels of abstraction can be extracted from databases of known structures, with the help of inverse Boltzmann statistics converted into empirical energy potentials.

According to the law of Boltzmann [137, 187] a particular state x of a physical system in equilibrium is occupied with probability $f(x)$:

$$f(x) = \frac{1}{Z} \mathrm{e}^{-\frac{E(x)}{kT}}, \quad \text{where} \quad Z = \int \cdots \int \mathrm{e}^{-\frac{E(x)}{kT}} \mathrm{d}x,$$

k is the Boltzmann's constant, T is the absolute temperature and Z is called the partition function. In a discrete state space, the integrals are replaced by sums. If the energies of all states x are known, the probability densities $f(x)$ can be computed.

Conversely, if the probability density functions f of a system are given [138], the energy can be calculated as:

$$E(x) = -kT \ln(f(x)) - kT \ln(Z).$$

This is frequently referred to as the inverse Boltzmann principle. Z cannot be evaluated by measuring the densities and therefore the energy can be only determined up to the additive constant $-kT \ln(Z)$.

Originally, Boltzmann's equation assumes that the system is in equilibrium. Also, the principle can only be applied to complete systems not to their parts. However, Finkelstein and coworkers showed that a Boltzmann-like distribution arises naturally from low energy conformations of random heteropolymers and similarly of proteins [38, 187]. This suggests that the Boltzmann model can be applied to derive empirical energy functions from ensembles of protein structures even though they are not systems in equilibrium.

4.2 Threading Using Pair-interaction Potentials

Different types of potentials are used in threading. The more accurate potentials rely on pair-interactions of residues (many-body interactions are not considered at all in threading due to their complexity). Most often, distances between pairs of residues, considering their amino acid side-chain types, are condensed into pair-interaction potentials using knowledge-based inverse Boltzmann approaches. Commonly used choices of interaction centers are the C_α atoms, the C_β atoms, the side-chain centers of mass, specially defined interaction centers or any side-chain atom [142].

Finding the globally optimum threading involving a pair-interaction scoring function is NP-hard if variable-length gaps and interactions between neighboring amino residues are allowed [77]. This means that, in order to find an optimal solution, an algorithm requires an amount of time that, in the worst case, is exponential in the size of the protein. Several strategies have been developed to tackle this.

Jones and coworkers [62] use a double dynamic programming algorithm in conjunction with potentials that do not require explicit modeling of all side-chain atoms. They use a set of knowledge-based potentials which are derived from a statistical analysis of known protein structures, according to the inverse Boltzmann principle [137]. For a given pair of atoms, a given residue sequence separation and a given interaction distance these potentials provide a measure of pseudo-energy, which relates to the probability of observing the proposed interaction in native protein structures. By providing different empirical potentials for different ranges of sequence separation, specific structural significance is conferred on each range. The short-range terms predominate in the matching of secondary structural elements. By threading a sequence segment onto the template of an α-helical conformation and evaluating the short-range potential terms, the probability of the sequence folding into an α-helix is evaluated. In a similar way, medium-range terms mediate the matching of super-secondary structural motifs, and the long-range terms the tertiary packing. Around each residue in turn, their algorithm uses dynamic programming as in sequence alignment to optimize the threading of the sequence onto the structure. It finally computes the best threading through the whole structure by means of a shortest-path algorithm.

Lathrop and Smith use core structural models to derive a branch and bound algorithm for threading [78]. A core structural model consists of several core segments. Each position of each core element is occupied by a single amino acid residue from the threaded sequence. Typically core segments correspond to secondary structure elements, i.e. helices or strands. They are connected by a set of loop regions. Neighboring positions are computed, where positions are defined to be neighbors if they contribute a pair-interaction term to the energy-score function. This often but not always requires that they lie close in space, that is make a noncovalent interaction. The use of core elements as larger building blocks reduces the problem size drastically.

Recursive dynamic programming (RDP) [154] is another approach to addressing the full threading problem using heuristics and without restriction to core elements. It is based on the "divide-and-conquer" paradigm and maps the target sequence onto the known backbone structure of a template protein in a stepwise fashion – a technique that is similar to computing local alignments but utilizing different cost functions. It starts by mapping parts of the target onto the template that show statistically significant similarity with the template sequence. After mapping, the template structure is modified in order to account for the mapped target residues. Then significant similarities between the yet unmapped parts of the target and the modified template are searched, and the resulting segments of the target are mapped onto the template. This recursive process of identifying segments in the target to be mapped onto the template and modifying the template is continued until no significant similarities between the remaining parts of target and template are found. Those parts which are left unmapped by the procedure are interpreted as gaps. The RDP method is robust in the sense that different local alignment methods can be used, several alternatives of mapping parts of the target onto the template can be handled and compared in the process, and the cost functions can be dynamically adapted to biological needs.

Xu and coworkers, in their RAPTOR (RApid Protein Threading by Operation Research technique) method, use a linear programming approach to do protein 3-D structure prediction via threading [181–183]. Based on the contact map graph of the protein 3-D structure template, the protein threading problem is formulated as a large-scale integer programming problem. This formulation is then relaxed to a linear programming problem, and solved by a branch-and-bound method. The final solution is globally optimal with respect to their energy functions. The energy function includes pairwise interaction preferences and allows variable gaps.

4.3 Threading using Frozen Approximation Algorithms

The alternative to using full pair-potentials in threading is to evaluate a target sequence with respect to the template structure's original native sequence. While threading the target onto the structure, the interaction partners in the potentials or a set of local environmental preferences are taken from the template protein. With these frozen approximation approaches [47,142], a globally optimum threading – of a problem with reduced complexity compared to the threading problem using full pair-interaction potentials – can be found using dynamic programming methods.

Bowie and coworkers [15] start with a known template structure and describe the environments of its residues by three types of properties: (i) the area of the residues buried in the protein and inaccessible to solvent, (ii) the fraction of side-chain area that is covered by polar atoms, and (iii) the local secondary structure. Based on these parameters each residue is categorized into an environment class. In this manner, a 3-D protein structure is converted into a 1-D string, like a sequence, which represents the environment class of each residue in the folded protein structure. With a sequence-alignment-like algorithm they then seek the most favorable alignment of a protein sequence to this environment string. An alignment column now aligns a residue in the target sequence with an environment class in the template structure. Using inverse Boltzmann statistics, knowledge-based scoring functions have been derived for this kind of match. Later the method was extended to also incorporate secondary structure predictions (helix, strand, coil) on the sequence side to be matched to the secondary structure of the templates [121].

Flöckner and Sippl base their method to determine a sequence to structure alignment on the Needleman–Wunsch algorithm [43,140]. While in sequence comparison the similarity of amino acids is measured directly, Flöckner and

Sippl evaluate the fitness of an amino acid of the query sequence within the template structure by using the energy field generated by the original template structure, while mutating that single residue to the type observed within the query sequence. For this mutated structure a knowledge-based energy function, composed of pairwise atom–atom interactions is evaluated for C_β–C_β interactions.

Also starting from known structures, Alexandrov and Zimmer [2] describe the environments of residues by counting the number of contacts that each amino acid makes in a structure. This information can be matched with sequence information by previously counting the so-called contact capacities, i.e. is the normalized number of contacts that each type of amino acid makes in an ensemble of proteins. Given the number of contact counts per sequence position in the structure and the number of counts a type of amino acid prefers, for each position in the structure's sequence certain types of amino acids are preferable. This information can be aligned to the target sequence with dynamic programming just as in sequence alignment.

5 Hybrid Methods and Recent Developments

5.1 Using Different Sources of Information

Secondary structure prediction and disorder prediction are methods for predicting additional structural features of amino acid sequences (see Chapter 9). Such methods can be used as stand-alone tools to learn more about a target sequence. Alternatively, they can be directly integrated into template identification methods by incorporating the additional sources of information into the scoring functions. In the context of this section, the focus is on the integration of these methods into sequence comparison.

5.1.1 Incorporating Secondary Structure Prediction into Frequency Profiles and HMMs

Often, one of the first steps in structure prediction is to predict the secondary structure of the target protein, that is to annotate each of a sequence's residues with a probability of being contained in a helix, coil or sheet (see Chapter 9). As a result, the field of secondary structure prediction has received considerable attention and is reaching a mature state [1,61,83,92,179].

Predicted secondary structure can be incorporated into other fold recognition methods, e.g. profile–profile, HMM or threading methods [39,88]. In the frequency profile case, the profile matrix with 20 columns for each type of amino acid is extended with an additional profile matrix with three columns for helix, coil and sheet. Both profile matrices are scored against other profile matrices of the same type and the results is merged to a joint score [46, 166]. Wang and Dunbrack state that incorporation of secondary structure information improves alignment accuracy slightly [171] and improves the search capabilities of the average score mentioned in Section 3.2.5 significantly.

Secondary structure information was also used to extend the HMM principle [64, 67]. Karchin and Karplus incorporated predicted local structure into so-called two-track profile HMMs. They did not rely on a simple helix–strand–coil definition of secondary structure, but experimented with a variety of local structure descriptions, and established which descriptions are most useful for improving fold recognition and alignment quality. On a test set of 1298 nonhomologous proteins, HMMs incorporating a three-letter alphabet improved fold recognition accuracy by 15% over HMMs using amino acids only. Comparing two-track HMMs to HMMs operating on amino acids only, on a difficult alignment test set of 200 protein pairs, Karchin found that HMMs with a six-letter secondary track improved alignment quality by 62%, relative to DALI [53] structural alignments.

5.1.2 Intrinsically Disordered Regions in Proteins

While the analysis and prediction of secondary structure is a matured subject, the interest in intrinsic disorder of proteins (see Chapter 9) has grown tremendously over the last couple of years [33, 63, 84, 102, 104, 119, 157, 158, 174].

Intrinsically disordered proteins do not fold into stable 3-D structure; instead, in solution they exist as an ensemble of interchanging conformations. There are examples of proteins which are disordered completely and others where only part of the amino acid sequence does not fold stably. In a recent study [103] based on six genomes, roughly 5% of proteins in bacteria, 7% in archea and 25% in eukaryotes were estimated to be mostly disordered. Ward and coworkers [174] predict an average of 2% of archaean, 4% of eubacteria and 33% of eukaryotic proteins to contain more than 30 residues of disorder. Thus, proteins with disordered regions seem to be especially common in eukaryotic cells. Often, proteins with disordered regions perform a function: they are involved in protein–protein [76, 82, 85] or protein–nucleic acid [148] interactions. Under certain conditions disordered fragments become structured during the process of interaction. In the electron density maps of X-ray crystallographic studies, disordered regions frequently do not appear; in NMR experiments, they appear highly flexible.

As stretches of amino acids that do not fold into a stable structure should not be predicted to have a structure, disorder prediction (see Chapter 9) has become important in structure prediction. Methods have been developed which annotate each residue of an amino acid sequence with a value of predicted disorder. These methods are based on machine learning techniques like neural nets or support vector machines and employ training sets of disordered regions to learn to discriminate ordered from disordered stretches of sequence [63, 84, 103, 104, 119, 168].

To the best of our knowledge such methods have not yet been incorporated into fold recognition methods as has been done with secondary structure prediction, but this should only be a matter of time.

5.1.3 Incorporating 3-D Structure into Frequency Profiles

Frequency profiles can be derived from sequences alone, as reviewed in Section 3.2.2. Alternatively, structure-based multiple alignments can be used for the generation of profiles. For this, multiple structure superposition is performed on the available structures to create a multiple alignment of their sequences which is then used to generate a frequency profile as described in Section 3.2.2 above [23, 69, 107, 114, 152, 184]. There are actually two aspects to this. First, using structure superposition to create multiple alignments even for cases where sequence alignment is not feasible. This potentially results in an increase of coverage at a loss of precision. Second, using available structures to generate more reliable seed alignments, increasing precision.

5.2 Combining Information

Recent contributions to the field of fold recognition have been integrative, collecting information from many sources. The GenTHREADER program [60, 93] for automatic fold recognition consists of a neural net which was trained to combine sequence alignment score, length information and energy potentials derived from threading into a single score representing the relationship between two proteins [60]. An improved version incorporates PSI-BLAST searches and also makes use of predicted secondary structure [93].

Another competitive example for the combination of information is the TASSER/PROSPECTOR suite of programs developed by Skolnick and coworkers. PROSPECTOR_3 is an iterative approach to search for diverse templates using a variety of pair potentials and scoring functions [143, 188]. Different frequency profiles are used in a first round to identify templates. These targets are further evaluated in subsequent evaluations of pair interactions. The scoring functions include a quasi-chemical based pair potential [141], a protein-specific, orientation-independent pair potential based on local sequence fragments [144] and a pair potential that depends on the orientation of the side-chains [144]. With the templates identified by PROSPECTOR, the TASSER program performs tertiary structure assembly via the rearrangement of continuous template fragments [189].

Another example is the SPARKS method (Sequence, secondary structure Profiles and Residue-level Knowledge-based energy Score) [191], in which Zhou and Zhou use a knowledge-based energy function for fold recognition.

Being a residue-level frozen approximation potential, the dynamic programming method can be used for alignment optimization. The potential contains a backbone torsion term, a buried surface term and a contact-energy term. With sequence profile and secondary structure information it is combined into a joint fold recognition method. Taking advantage of sequence and structure methods it was highly competitive in the latest CASP experiment.

Like this one, there are a number of other approaches integration information from programs running locally or collecting information from web services [166].

5.3 Meta-servers

Meta-servers collect and analyze results from individual web servers and combine them into a joint result. Benchmarking results obtained in the last years indicate that, on average, meta-servers are more accurate than individual methods.

One first successful attempt to benefit from a number of distributed information sources was the PCONS meta-server, which concentrates on a number of reliable servers at different locations and selects the most abundant fold among their high-scoring models [86]. It translates the confidence scores reported by each server into uniformly scaled values corresponding to the expected accuracy of each model. The translated scores as well as the similarity between initial models is used as input to a neural network in charge of the final selection.

Several other meta-methods followed soon afterwards. Current methods differ in how the initial models are compared, whether scores provided by the individual methods are used and how the final model is generated.

The 3D-Jury system uses the rationale that the high-scoring models which are produced by several servers are closer to the native structure than the single model with highest score. Thus, models occurring with higher than expected frequencies are are taken for the preferred conformation [44, 163].

The 3D-SHOTGUN meta-predictor employs techniques of so-called cooperative algorithms from computer vision. As input it takes the models with their confidence scores. The result is a hybrid model, which is spliced from fragments of the input models. It has the potential of covering a larger part of the native protein than any template structure alone. Thus, 3D-SHOTGUN entails the first fold recognition meta-predictor attempt to go beyond the simple selection of one of the input models [40].

6 Assessment of Models

Once a template is identified it can be assessed in a number of ways. (i) The significance of the selection of the template can be estimated. (ii) After a model has been constructed on the basis of the template, the quality of the 3-D model can be scored. (iii) If the native structure for which the prediction was made becomes known later, the quality of the model can be evaluated, in terms of how faithfully it represents the true structure. Thus, conclusions about the method producing the prediction can be drawn.

6.1 Estimating Significance of Sequence Hits

Essentially, methods for template identification compute a list of candidate templates for a target sequence. The method typically employs a scoring system or (virtual) energy function according to which the list is sorted and the maximum scoring candidate is selected as template. Before using a template to build a model the question how reliably the template is related to the target needs to be addressed.

The classic approach to this is to calculate the probability of obtaining a maximum score greater than the observed score assuming that the protein sequences compared with the scoring system are unrelated. This probability is called p-value. To compute it one needs to know the distribution of maximal scores for unrelated sequences for the particular scoring scheme. The E-value is a similar concept, additionally taking the size of the database of templates into account [5,6].

For some template identification methods the score distributions are known, thus scores can be readily converted into p-values (see Section 4 in Chapter 3). For other methods empirical confidence scores have been developed.

For optimal local gapless sequence alignments of independent random sequences the score distributions are known to be of an asymptotically extreme-value or Gumbel form [65]: $P(score > t) \approx 1 - \mathrm{e}^{-K\mathrm{e}^{-\lambda t}}$, where λ depends only on the scoring system and K depends on the scoring system and the sequence lengths, such that the distribution reflects the fact that the chance of spurious high scores increases with sequence lengths. The dependence of the parameters on the scoring system and sequence lengths is known.

For local alignments with gaps of unrelated biological sequences no general theory is available. However, there is considerable evidence that the distribution is still of extreme-value form and parameters can be fitted experimentally [4,75,80,95,112,175]. Local alignments using frequency profile sequences were also shown to follow an extreme-value distribution [95].

For optimal global alignments, whether with plain sequences or sequence profiles, theoretically neither the family of distributions nor the dependence of the expected score (or of other parameters) on the sequence lengths is known. However, there exist approaches for experimentally fitting distributions to scores [109, 176].

For frozen approximation sequence–structure alignment the situation is similar to sequence alignment: the local threading scores of sample sequence–structure pairs often can be fitted to a Gumbel distribution [147]. In the global threading case the distribution is unknown.

For the methods for which no theory is available, heuristic approaches exist for estimating reliability of the predictions. One generic approach is to derive a function of target and template which is reasonably associated with reliability, and then statistically test it on a set of proteins to estimate the probability for an identified target being correct (e.g. Ref. [147]). One such target–template comparison function is, for example, the difference of the score of the target sequence aligned to the template and the reversed target sequence aligned to the template. This is much faster than repeatedly evaluating the score relative to that of a randomized sequence [66, 143].

For a more detailed discussion, see Ref. [34], where the classical statistical assessment of significance of scores is also compared to Bayesian approaches.

6.2 Scoring 3-D Model Quality: Model Quality Assessment Programs (MQAPs)

The template identification methods discussed above compute suggestions for templates. With a target-template alignment, a model can be computed. MQAPs serve to distinguish near-native structures (i.e. "good" predictions) from decoys (i.e. "bad" predictions). MQAPs are programs that receive as input a 3-D model of a protein structure and produce as output a real number representing the quality of the model (http://www.cs.bgu.ac.il/∼dfischer/CAFASP4/mqap.html). MQAPs only use the model as input, not the native structure, and thus stand in contrast to methods that evaluate the quality of a model when comparing it to the native structure.

In contrast to scoring functions in sequence–structure alignment and to physical energy functions, MQAPs operate on an intermediate level – they are more flexible than a sequence–structure alignment function as the dynamic programming paradigm used in alignment imposes the requirement of prefix optimality which is not required in MQAPs. On the other hand, MQAP functions are not sensitive to ruptures in the sequences in contrast to physical energy functions. MQAPs target at scoring the quality of predicted models. Typically, MQAPs use one or more different statistical potentials, representing information coded in protein structures [87, 116, 139, 159]. For example, the FRST method uses a combination of pairwise, solvation, hydrogen bond and torsion angle potentials [159]. An intuitive test for a scoring function is picking the native structure among a set of decoys. A number of decoy sets have been made available [127] and are used for training.

In 2004, CAFASP4 [42] provided infrastructure to perform a fully automated blind test of MQAPs on the CASP target proteins and the structure predictions made during CASP. In this test the five following programs were rated as most reliable: FRST [159], Verify3-D [87], RAPDF [128], ProsaII [139] and ProQ [169].

Generating different alignments between target and template using various alignment methods and then employing an MQAP to pick out the best one can potentially lead to an improved overall alignment result.

6.3 Evaluation of Protein Structure Prediction: Critical Assessment of Techniques for Protein Structure Prediction

The performance of methods like the ones reviewed above needs assessment. The performance of sequence–structure alignment methods can be assessed either by testing their fold recognition performance [20, 37, 71] or by benchmarking their alignment quality [130, 171]. For assessing fold recognition performance, classification performance of different methods is tested versus a standard like the SCOP database for structure classification. For measuring alignment quality, the sequence–structure alignments produced by threading methods are compared to high-quality alignments as produced by structure superposition methods. CASP, a blind testing experiment, has had a large impact on the community and is therefore summarized in the following.

CASP was started in 1994 [98]. The idea was to establish a clearinghouse between experimental and predicted protein structure. Protein sequences whose structure is currently being analyzed experimentally are made available to structure prediction groups. Structural models are predicted by a number of participating groups and submitted to the CASP organizers before the release of the crystal structures. After the release of the crystal structures, the predicted models are compared with them by a number of human experts – the CASP assessors. Typically, the structures are categorized according to their difficulty into homology modeling, fold recognition and *de novo* targets. CASP has been held biannually. The number of targets has been growing from 33 (CASP1 in 1994) to 42, (CASP2), 43 (CASP3), 43 (CASP4), 67 (CASP5) and 64 (CASP6 in 2004), as well as the number of participating prediction groups from 35 (CASP1) to 152 (CASP2) to 201 human groups and 65 servers in CASP6.

Within the original setup of CASP humans can interact with computer programs to generate protein models. In this setting, it is hard to discriminate between the contribution of the human and the program, respectively. This issue was resolved by introducing CAFASP [41], where the additional FA in the acronym stands for fully automated. In CAFASP, server programs are directly contacted by the clearinghouse and have to respond within a short timeframe (usually 48 h) and without human intervention.

In order to perform the comparisons of predicted models and experimental structures, different assessors have chosen different approaches. Some completely relied on manual inspection (which is a tremendous amount of work). Due to the number of participating groups, computerized preprocessing of the results is becoming the standard procedure. Different distance measures for measuring the fractional similarity between the experimental and the predicted structure have been employed. Currently the GDT_TS score is used for CASP and the MaxSub score in CAFASP. The GDT_TS score is computed by the LGA program which simultaneously optimizes for local structure and global RMSD superposition using different cutoffs [186]. MaxSub aims at identifying the largest subset of C_α atoms of a model that superimpose "well" over the experimental structure [135]. Both measures produce a single normalized score that represents the quality of the model. Both measures differ in details, but correlate on concrete examples [190].

CASP5 (2002) results for fold recognition methods are summarized in Ref. [70], results of CASP6 (2004) are in press at the time of writing [96, 161, 172]. While there is an obvious danger of overtraining to the experiment, the CASP community has been eager to pick up new trends and find ways to evaluate them. Examples are the evaluation of predicted intrinsic disorder at CASP5 and CASP6 or the assessment of prediction of domain boundaries and model quality assessment programs at CASP6.

Continuous experiments, LiveBench and EVA, are performed by weekly extracting newly published structures from the PDB and submitting them to automated servers. Based on automated measures like MaxSub, the quality of the predictions of participating servers can be measured online [21, 122].

7 Programs and Web Resources

The web resources for protein structure prediction are extensive. Since search engines have a tendency to be more up-to-date and practical than link lists in books, we give an overview and further pointers here only. Good starting points for fold recognition via the Internet are meta-servers and the CASP experiment. Meta-servers provide functionality for unproblematically exercising a number of servers simultaneously in order to perform structure predictions. The CASP experiment provides a list of the best performing methods today; although some of these might not be as easily accessible or very compute intensive to exercise.

Bioinfo/3D-Jury	http://bioinfo.pl/meta
PredictProtein	http://www.embl-heidelberg.de/predictprotein/predictprotein
CASP	http://predictioncenter.org
CAFASP	http://www.cs.bgu.ac.il/∼dfischer/CAFASP4

For high-throughput experiments as well as confidentiality reasons one may want to install fold recognition software locally. Unfortunately, this software tends to depend on a number of libraries (templates, profiles, potentials, motifs, etc.) which need to be up-to-date in order to be performant and which have a tendency to be cumbersome to install. An obvious starting point for downloadable software is PSI-BLAST. Some HMM software is freely available. Most profile software is available through contacting the respective authors only. URLs of BLAST and two HMM libraries are:

BLAST	http://www.ncbi.nlm.nih.gov/BLAST
HMMer	http://hmmer.wustl.edu
SAM	http://www.cse.ucsc.edu/compbio/sam.html

Links to key databases are:

PDB	http://www.rcsb.org/pdb
SCOP	http://scop.mrc-lmb.cam.ac.uk/scop
Entrez DB Collection	http://www.ncbi.nlm.nih.gov/gquery/gquery.fcgi

Also Kevin Karplus provides an up to date link list, related to this chapter:

http://www.soe.ucsc.edu/∼karplus/compbio_pages

Acknowledgments

This work was funded in part by grant Le 491/14 of the Deutsche Forschungsgemeinschaft. I thank my colleagues at the Max-Planck-Institut for Informatics for their support. I am particularly grateful to Mario Albrecht, Francisco S. Domingues, Andreas Kaemper, Lars Kunert, Oliver Sander and Thomas Lengauer for critically reading the manuscript and improving it considerably.

References

1 Albrecht, M., S. C. E. Tosatto, T. Lengauer, and G. Valle. 2003. Simple consensus procedures are effective and sufficient in secondary structure prediction. Protein Eng. **16**: 459–62.

2 Alexandrov, N., R. Nussinov, and R. Zimmer. 1996. Fast protein fold recognition via sequence to structure alignment and contact capacity potentials. Pac. Symp. Biocomput. **1**: 53–72.

3 Altschul, S., R. Caroll, and D. Lipman. 1989. Weights for data related by a tree. J. Mol. Biol. **207**: 647–53.

4 Altschul, S. and W. Gish. 1996. Local alignment statistics. Methods Enzymol. **266**: 460–80.

5 Altschul, S. F., W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. 1990. Basic local alignment search tool. J. Mol. Biol. **215**: 403–10.

6 ALTSCHUL, S. F., T. L. MADDEN, A. A. SCHÄFFER, J. ZHANG, Z. ZHANG, W. MILLER, AND D. J. LIPMAN. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**: 3389–402.

7 ANFINSEN, C. 1973. Principles that govern the folding of protein chains. Science **181**: 223–30.

8 ANFINSEN, C. B., E. HABER, M. SELA, AND F. H. WHITE. 1961. The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. Proc. Natl. Acad. Sci. USA **47**: 1309–14.

9 BALDI, P. AND S. BRUNAK. 2001. *Bioinformatics: The Machine Learning Approach*. MIT Press, Cambridge, MA.

10 BALDI, P. AND Y. CHAUVIN. 1994. Smooth on-line learning algorithms for hidden markov models. Neural Comput. **6**: 305–16.

11 BATEMAN, A., L. COIN, R. DURBIN, ET AL. 2004. The Pfam protein families database. Nucleic Acids Res. **32**: D138–41.

12 BEN-HUR, A. AND D. BRUTLAG. 2003. Remote homology detection: a motif based approach. Bioinformatics **19 (Suppl. 1)**: i26–33.

13 BERG, J., J. TYMOCZKO, AND L. STRYER. 2001. *Biochemistry*. Freeman, San Francisco, CA.

14 BOURNE, P. E. AND H. WEISSIG. 2003. *Structural Bioinformatics*. Wiley, New York, NY.

15 BOWIE, J., R. LUTHY, AND D. EISENBERG. 1991. A method to identify protein sequences that fold into a known three-dimensional structure. Science **253**: 164–70.

16 BRENNER, S., P. KOEHL, AND M. LEVITT. 2000. The ASTRAL compendium for protein structure and sequence analysis. Nucleic Acids Res. **28**: 254–6.

17 BROWN, M. P., R. HUGHEY, A. KROGH, I. S. MIAN, K. SJÖLANDER, AND D. HAUSSLER. 1993. Using Dirichlet mixture priors to derive hidden Markov models for protein families. Proc. ISMB **1**: 47–55.

18 BUCHER, P. AND A. BAIROCH. 1994. A generalized profile syntax for biomolecular sequence motifs and its function in automatic sequence interpretation. Proc. ISMB **2**: 53–61.

19 BUCHETE, N.-V., J. E. STRAUB, AND D. THIRUMALAI. 2004. Development of novel statistical potentials for protein fold recognition. Curr. Opin. Struct. Biol. **14**: 225–32.

20 BUJNICKI, J., A. ELOFSSON, D. FISCHER, AND L. RYCHLEWSKI. (2001a). LiveBench-2: large-scale automated evaluation of protein structure prediction servers. Proteins **45 (Suppl. 5)**: 184–91.

21 BUJNICKI, J. M., A. ELOFSSON, D. FISCHER, AND L. RYCHLEWSKI. (2001b). LiveBench-1: continuous benchmarking of protein structure prediction servers. Protein Sci. **10**: 352–61.

22 CARILLO, H. AND D. LIPMAN. 1988. The multiple sequence alignment in biology. SIAM J. Appl. Math. **48**: 1073–82.

23 CASBON, J. AND M. A. S. SAQI. 2005. S4: structure-based sequence alignments of SCOP superfamilies. Nucleic Acids Res. **33**: D219–22.

24 CHANDONIA, J.-M., G. HON, N. S. WALKER, L. L. CONTE, P. KOEHL, M. LEVITT, AND S. E. BRENNER. 2004. The ASTRAL Compendium in 2004. Nucleic Acids Res. **32**: D189–92.

25 CHOO, K. H., J. C. TONG, AND L. ZHANG. 2004. Recent applications of Hidden Markov Models in computational biology. Genomics Proteomics Bioinf. **2**: 84–96.

26 CHURCHILL, G. A. 1989. Stochastic models for heterogeneous DNA sequences. Bull. Math. Biol. **51**: 79–94.

27 CLAVERIE, J.-M. 1994. Some useful statistical properties of position-weight matrices. Comput. Chem. **18**: 287–94.

28 CROOKS, G. E., G. HON, J.-M. CHANDONIA, AND S. E. BRENNER. 2004. WebLogo: a sequence logo generator. Genome Res. **14**: 1188–90.

29 DAYHOFF, M. O., R. M. SCHWARTZ, AND B. C. ORCUTT. 1978. A model of evolutionary change in proteins. In DAYHOFF M. O. (ed.) *Atlas of Protein Sequence and Structure*, Volume 5,

Chapter 22, pp. 345–352. National Biomedical Research Foundation.

30 Dempster, A., N. Laird, and D. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. J. R. Stat. Soc. B **39**: 1–38.

31 Dinner, A. R., A. Sali, L. J. Smith, C. M. Dobson, and M. Karplus. 2000. Understanding protein folding via free-energy surfaces from theory and experiment. Trends Biochem. Sci. **25**: 331–9.

32 Dunbrack, R. 1999. Comparative modeling of CASP3 targets using PSI-BLAST and SCWRL. Proteins **37 (Suppl. 3)**: 81–7.

33 Dunker, A. K. and Z. Obradovic. 2001. The protein trinity – linking function and disorder. Nat. Biotechnol. **19**: 805–6.

34 Durbin, R., S. Eddy, A. Krogh, and G. Mitchison. 1998. *Biological Sequence Analysis*. Cambridge University Press, Cambridge.

35 Eddy, S. R. 1998. Profile hidden Markov models. Bioinformatics **14**: 755–63.

36 Edgar, R. C. and K. Sjölander. 2004. COACH: profile–profile alignment of protein families using hidden Markov models. Bioinformatics **20**: 1309–18.

37 Eyrich, V. A., D. Przybylski, I. Y. Y. Koh, O. Grana, F. Pazos, A. Valencia, and B. Rost. 2003. CAFASP3 in the spotlight of EVA. Proteins **53 (Suppl. 6)**: 548–60.

38 Finkelstein, A. V., A. Y. Badretdinov, and A. M. Gutin. 1995. Why do protein architectures have Boltzmann-like statistics? Proteins **23**: 142–50.

39 Fischel-Ghodsian, F., G. Mathiowitz, and T. Smith. 1990. Alignment of protein sequences using secondary structure: a modified dynamic programming method. Protein Eng. **3**: 577–81.

40 Fischer, D. 2003. 3D-SHOTGUN: a novel, cooperative, fold-recognition meta-predictor. Proteins **51**: 434–41.

41 Fischer, D., C. Barret, K. Bryson, et al. 1999. CAFASP-1: critical assessment of fully automated structure prediction methods. Proteins **37 (Suppl. 3)**: 209–17.

42 Fischer, D., L. Rychlewski, R. L. Dunbrack, A. R. Ortiz, and A. Elofsson. 2003. CAFASP3: the third critical assessment of fully automated structure prediction methods. Proteins **53 Suppl. 6**: 503–16.

43 Flöckner, H., M. Braxenthaler, P. Lackner, M. Jaritz, M. Ortner, and M. J. Sippl. 1995. Progress in fold recognition. Proteins **23**: 376–86.

44 Ginalski, K., A. Elofsson, D. Fischer, and L. Rychlewski. 2003. 3D-Jury: a simple approach to improve protein structure predictions. Bioinformatics **19**: 1015–8.

45 Ginalski, K., N. V. Grishin, A. Godzik, and L. Rychlewski. 2005. Practical lessons from protein structure prediction. Nucleic Acids Res. **33**: 1874–91.

46 Ginalski, K., J. Pas, L. S. Wyrwicz, M. von Grotthuss, J. M. Bujnicki, and L. Rychlewski. 2003. ORFeus: detection of distant homology using sequence profiles and predicted secondary structure. Nucleic Acids Res. **31**: 3804–7.

47 Godzik, A., A. Kolinski, and J. Skolnick. 1992. Topology fingerprint approach to the inverse protein folding problem. J. Mol. Biol. **227**: 227–38.

48 Gotoh, O. 1982. An improved algorithm for matching biological sequences. J. Mol. Biol. **162**: 705–8.

49 Gribskov, M., A. McLachlan, and D. Eisenberg. 1987. Profile analysis: detection of distantly related proteins. Proc. Natl Acad. Sci. USA **84**: 4355–8.

50 Gribskov, M. and S. Veretnik. 1996. Identification of sequence patterns with profile analysis. Methods Enzymol. **266**: 198–212.

51 Henikoff, S. and J. G. Henikoff. 1992. Amino acid substitution matrices from protein blocks. Proc. Natl Acad. Sci. USA **89**: 10 915–9.

52 Henikoff, S. and J. G. Henikoff. 1994. Position-based sequence weights. J. Mol. Biol. **243**: 574–8.

53 Holm, L. and C. Sander. 1992. Evaluation of protein models by atomic solvation preference. J. Mol. Biol. **225**: 93–105.

54 Holm, L. and C. Sander. 1996. Mapping the protein universe. Science **273**: 595–603.

55 Hooft, R. W., C. Sander, M. Scharf, and G. Vriend. 1996. The PDBFINDER database: a summary of PDB, DSSP and HSSP information with added value. Comput. Appl. Biosci. **12**: 525–9.

56 Jaakkola, T., M. Diekhans, and D. Haussler. 1999. Using the Fisher kernel method to detect remote protein homologies. Proc. ISMB **7**: 149–58.

57 Jaakkola, T., M. Diekhans, and D. Haussler. 2000. A discriminative framework for detecting remote protein homologies. J. Comput. Biol. **7**: 95–114.

58 Jones, D., R. Miller, and J. Thornton. 1995. Successful protein fold recognition by optimal sequence threading validated by rigorous blind testing. Proteins **23**: 387–97.

59 Jones, D. T. 1997. Progress in protein structure prediction. Curr. Opin. Struct. Biol. **7**: 377–87.

60 Jones, D. T. (1999a). GenTHREADER: an efficient and reliable protein fold recognition method for genomic sequences. J. Mol. Biol. **287**: 797–815.

61 Jones, D. T. (1999b). Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. **292**: 195–202.

62 Jones, D. T., W. R. Taylor, and J. M. Thornton. 1992. A new approach to protein fold recognition. Nature **358**: 86–9.

63 Jones, D. T. and J. J. Ward. 2003. Prediction of disordered regions in proteins from position specific score matrices. Proteins **53 (Suppl. 6)**: 573–8.

64 Karchin, R., M. Cline, Y. Mandel-Gutfreund, and K. Karplus. 2003. Hidden Markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. Proteins **51**: 504–14.

65 Karlin, S. and S. Altschul. 1990. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. Proc. Natl Acad. Sci. USA **87**: 2264–8.

66 Karplus, K., C. Barrett, and R. Hughey. 1998. Hidden Markov models for detecting remote protein homologies. Bioinformatics **14**: 846–56.

67 Karplus, K., R. Karchin, J. Draper, J. Casper, Y. Mandel-Gutfreund, M. Diekhans, and R. Hughey. 2003. Combining local-structure, fold-recognition, and new fold methods for protein structure prediction. Proteins **53 (Suppl. 6)**: 491–6.

68 Karplus, K., K. Sjölander, C. Barrett, M. Cline, D. Haussler, R. Hughey, L. Holm, and C. Sander. 1997. Predicting protein structure using hidden Markov models. Proteins **29 (Suppl. 1)**: 134–9.

69 Kelley, L., R. MacCallum, and M. Sternberg. 2000. Enhanced genome annotation using structural profiles in the program 3D-PSSM. J. Mol. Biol. **299**: 499–520.

70 Kinch, L. N., J. O. Wrabl, S. S. Krishna, et al. 2003. CASP5 assessment of fold recognition target predictions. Proteins **53 (Suppl. 6)**: 395–409.

71 Koh, I. Y. Y., V. A. Eyrich, M. A. Marti-Renom, et al. 2003. EVA: Evaluation of protein structure prediction servers. Nucleic Acids Res. **31**: 3311–5.

72 Koonin, E. V., Y. I. Wolf, and G. P. Karev. 2002. The structure of the protein universe and genome evolution. Nature **420**: 218–23.

73 Krogh, A., M. Brown, I. S. Mian, K. Sjölander, and D. Haussler. 1994. Hidden Markov models in computational biology. Applications to protein modeling. J. Mol. Biol. **235**: 1501–31.

74 Krogh, A. and G. Mitchison. 1995. Maximum entropy weighting of aligned sequences of protein or dna. Proc. ISMB **3**: 215–21.

75 Kschischo, M., M. Lässig, and Y.-K. Yu. 2005. Toward an accurate statistics of gapped alignments. Bull. Math. Biol. **67**: 169–91.

76 Lacy, E. R., I. Filippov, W. S. Lewis, S. Otieno, L. Xiao, S. Weiss, L. Hengst, and R. W. Kriwacki. 2004. p27 binds cyclin–CDK complexes through

a sequential mechanism involving binding-induced protein folding. Nat. Struct. Mol. Biol. **11**: 358–64.

77 LATHROP, R. H. 1994. The protein threading problem with sequence amino acid interaction preferences is NP-complete. Protein Eng. **7**: 1059–68.

78 LATHROP, R. H. AND T. F. SMITH. 1996. Global optimum protein threading with gapped alignment and empirical pair score functions. J. Mol. Biol. **255**: 641–65.

79 LESLIE, C. S., E. ESKIN, A. COHEN, J. WESTON, AND W. S. NOBLE. 2004. Mismatch string kernels for discriminative protein classification. Bioinformatics **20**: 467–76.

80 LEVITT, M. AND M. GERSTEIN. 1998. A unified statistical framework for sequence comparison and structure comparison. Proc. Natl Acad. Sci. USA **95**: 5913–20.

81 LIAO, L. AND W. S. NOBLE. 2003. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. J. Comput. Biol. **10**: 857–68.

82 LIDDINGTON, R. C. 2004. Structural basis of protein–protein interactions. Methods Mol. Biol. **261**: 3–14.

83 LIN, K., V. A. SIMOSSIS, W. R. TAYLOR, AND J. HERINGA. 2005. A simple and fast secondary structure prediction method using hidden neural networks. Bioinformatics **21**: 152–9.

84 LINDING, R., L. J. JENSEN, F. DIELLA, P. BORK, T. J. GIBSON, AND R. B. RUSSELL. 2003. Protein disorder prediction: implications for structural proteomics. Structure (Camb.) **11**: 1453–9.

85 LIU, D., R. ISHIMA, K. I. TONG, ET AL. 1998. Solution structure of a TBP–TAF(II)230 complex: protein mimicry of the minor groove surface of the TATA box unwound by TBP. Cell **94**: 573–83.

86 LUNDSTRÖM, J., L. RYCHLEWSKI, J. BUJNICKI, AND A. ELOFSSON. 2001. Pcons: a neural-network-based consensus predictor that improves fold recognition. Protein Sci. **10**: 2354–62.

87 LÜTHY, R., J. BOWIE, AND D. EISENBERG. 1992. Assessment of protein models with three-dimensional profiles. Nature **356**: 83–5.

88 LÜTHY, R., A. MCLACHLAN, AND D. EISENBERG. 1991. Secondary structure-based profiles: use of structure-conserving scoring tables in searching protein sequence databases for structural similarities. Proteins **10**: 229–39.

89 LÜTHY, R., I. XENARIOS, AND P. BUCHER. 1994. Improving the sensitivity of the sequence profile method. Protein Sci. **3**: 139–46.

90 LYNGSØ, R. B., C. N. PEDERSEN, AND H. NIELSEN. 1999. Metrics and similarity measures for hidden Markov models. Proc. ISMB **7**: 178–86.

91 MARTI-RENOM, M. A., M. MADHUSUDHAN, AND A. SALI. 2004. Alignment of protein sequences by their profiles. Protein Sci. **13**: 1071–87.

92 MCGUFFIN, L. J., K. BRYSON, AND D. T. JONES. 2000. The PSIPRED protein structure prediction server. Bioinformatics **16**: 404–5.

93 MCGUFFIN, L. J. AND D. T. JONES. 2003. Improvement of the GenTHREADER method for genomic fold recognition. Bioinformatics **19**: 874–81.

94 Morris, A. L., M. W. MacArthur, E. G. Hutchinson, and J. M. Thornton (1992, Apr). Stereochemical quality of protein structure coordinates. Proteins **12**: 345–64.

95 MOTT, R. 2000. Accurate formula for *p*-values of gapped local sequence and profile alignments. J. Mol. Biol. **300**: 649–59.

96 MOULT, J., K. FIDELIS, A. TRAMONTANO, B. ROST, AND T. HUBBARD. 2005. Critical assessment of methods of protein structure prediction (CASP) – round VI. Proteins **61 (Suppl. 7)**: 3–7.

97 MOULT, J., K. FIDELIS, A. ZEMLA, AND T. HUBBARD. 2003. Critical assessment of methods of protein structure prediction (CASP) – round V. Proteins **53 (Suppl. 6)**: 334–9.

98 MOULT, J., J. T. PEDERSEN, R. JUDSON, AND K. FIDELIS. 1995. A large-scale experiment to assess protein structure prediction methods. Proteins **23**: ii–v.

99 MURZIN, A. G., S. E. BRENNER, T. HUBBARD, AND C. CHOTHIA. 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. **247**: 536–40.

100 NEEDLEMAN, S. B. AND C. D. WUNSCH. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. **48**: 443–53.

101 NOTREDAME, C., D. HIGGINS, AND J. HERINGA. 2000. T-Coffee: A novel method for fast and accurate multiple sequence alignment. J. Mol. Biol. **302**: 205–17.

102 OBRADOVIC, Z., K. PENG, S. VUCETIC, P. RADIVOJAC, C. J. BROWN, AND A. K. DUNKER. 2003. Predicting intrinsic disorder from amino acid sequence. Proteins **53 (Suppl. 6)**: 566–72.

103 OLDFIELD, C. J., Y. CHENG, M. S. CORTESE, C. J. BROWN, V. N. UVERSKY, AND A. K. DUNKER. 2005. Comparing and combining predictors of mostly disordered proteins. Biochemistry **44**: 1989–2000.

104 OLDFIELD, C. J., E. L. ULRICH, Y. CHENG, A. K. DUNKER, AND J. L. MARKLEY. (2005). Addressing the intrinsic disorder bottleneck in structural proteomics. Proteins **59**: 444–53.

105 ONUCHIC, J. N. AND P. G. WOLYNES. 2004. Theory of protein folding. Curr. Opin. Struct. Biol. **14**: 70–5.

106 Orengo, C. A., A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. THORNTON. 1997. CATH – a hierarchic classification of protein domain structures. Structure **5**: 1093–108.

107 O'SULLIVAN, O., K. SUHRE, C. ABERGEL, D. G. HIGGINS, AND C. NOTREDAME. (2004). 3DCoffee: combining protein sequences and structures within multiple sequence alignments. J. Mol. Biol. **340**: 385–95.

108 PANCHENKO, A. R. 2003. Finding weak similarities between proteins by sequence profile comparison. Nucleic Acids Res. **31**: 683–9.

109 PANG, H., J. TANG, S.-S. CHEN, AND S. TAO. 2005. Statistical distributions of optimal global alignment scores of random protein sequences. BMC Bioinformatics **6**: 257.

110 PARK, J., K. KARPLUS, C. BARRETT, R. HUGHEY, D. HAUSSLER, T. HUBBARD, AND C. CHOTHIA. 1998. Sequence comparisons using multiple sequences detect three times as many remote homologues as pairwise methods. J. Mol. Biol. **284**: 1201–10.

111 PEARSON, W. 1996. Effective protein sequence comparison. Methods Enzymol. **266**: 227–58.

112 PEARSON, W. R. 1998. Empirical statistical estimates for sequence similarity searches. J. Mol. Biol. **276**: 71–84.

113 PEARSON, W. R. AND D. J. LIPMAN. 1988. Improved tools for biological sequence comparison. Proc. Natl Acad. Sci. USA **85**: 2444–8.

114 PETREY, D., Z. XIANG, C. L. TANG, ET AL. 2003. Using multiple structure alignments, fast model building, and energetic analysis in fold recognition and homology modeling. Proteins **53 (Suppl. 6)**: 430–5.

115 PETSKO, G. A. AND D. RINGE. 2003. *Protein Structure and Function*. New Science Press, London.

116 PETTITT, C. S., L. J. MCGUFFIN, AND D. T. JONES. 2005. Improving sequence-based fold recognition by using 3D model quality assessment. Bioinformatics **21**: 3509–15.

117 PIETROKOVSKI, S. 1996. Searching databases of conserved sequence regions by aligning protein multiple-alignments. Nucleic Acids Res. **24**: 3836–45.

118 RABINER, L. 1989. A tutorial on hidden markov models and selected applications in speech recognition. Proc. IEEE **77**: 257–86.

119 RADIVOJAC, P., Z. OBRADOVIC, D. K. SMITH, G. ZHU, S. VUCETIC, C. J. BROWN, J. D. LAWSON, AND A. K. DUNKER. 2004. Protein flexibility and intrinsic disorder. Protein Sci. **13**: 71–80.

120 RHODES, G. 2004. *Crystallography Made Crystal Clear*. Academic Press, New York, NY.

121 RICE, D. AND D. EISENBERG. 1997. A 3D-1D substitution matrix for protein fold recognition that includes predicted secondary structure of the sequence. J. Mol. Biol. **267**: 1026–38.

122 Rychlewski, L. and D. Fischer. 2005. LiveBench-8: the large-scale, continuous assessment of automated protein structure prediction. Protein Sci. **14**: 240–5.

123 Rychlewski, L., L. Jaroszewski, W. Li, and A. Godzik. 2000. Comparison of sequence profiles. Strategies for structural predictions using sequence information. Protein Sci. **9**: 232–41.

124 Rychlewski, L., B. Zhang, and A. Godzik. 1998. Fold and function predictions for *Mycoplasma genitalium* proteins. Fold. Design **3**: 229–38.

125 Sadreyev, R. and N. Grishin. 2003. COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance. J. Mol. Biol. **326**: 317–36.

126 Saigo, H., J.-P. Vert, N. Ueda, and T. Akutsu. 2004. Protein homology detection using string alignment kernels. Bioinformatics **20**: 1682–9.

127 Samudrala, R. and M. Levitt. 2000. Decoys 'R' Us: a database of incorrect conformations to improve protein structure prediction. Protein Sci. **9**: 1399–401.

128 Samudrala, R. and J. Moult. 1998. An all-atom distance-dependent conditional probability discriminatory function for protein structure prediction. J. Mol. Biol. **275**: 895–916.

129 Sánchez, R. and A. Sali. 1997. Evaluation of comparative protein structure modeling by MODELLER-3. Proteins **29 (Suppl. 1)**: 50–8.

130 Sauder, J., J. Arthur, and R. Dunbrack. 2000. Large-scale comparison of protein sequence alignment algorithms with structure alignments. Proteins **40**: 6–22.

131 Schäffer, A., Y. Wolf, C. Ponting, E. Koonin, L. Aravind, and S. Altschul. 1999. IMPALA: matching a protein sequence against a collection of PSI-BLAST-constructed position-specific score matrices. Bioinformatics **15**: 1000–11.

132 Schölkopf, B. and A. Smola. 2002. *Learning with Kernels*. MIT Press, Cambridge, MA.

133 Schwede, T., J. Kopp, N. Guex, and M. C. Peitsch. 2003. SWISS-MODEL: An automated protein homology-modeling server. Nucleic Acids Res. **31**: 3381–5.

134 Sibbald, P. and P. Argos. 1990. Weighting aligned protein or nucleic acid sequences to correct for unequal representation. J. Mol. Biol. **216**: 813–8.

135 Siew, N., A. Elofsson, L. Rychlewski, and D. Fischer. 2000. MaxSub: an automated measure for the assessment of protein structure prediction quality. Bioinformatics **16**: 776–85.

136 Simossis, V., J. Kleinjung, and J. Heringa. 2005. Homology-extended sequence alignment. Nucleic Acids Res. **33**: 816–24.

137 Sippl, M. 1990. Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. J. Mol. Biol. **213**: 859–83.

138 Sippl, M. (1993a). Boltzmann's principle, knowledge-based mean fields and protein folding. An approach to the computational determination of protein structures. J. Comput. Aided Mol. Des. **7**: 473–501.

139 Sippl, M. (1993b). Recognition of errors in three-dimensional structures of proteins. Proteins **17**: 355–62.

140 Sippl, M. J. and H. Flöckner. 1996. Threading thrills and threats. Structure **4**: 15–9.

141 Skolnick, J., L. Jaroszewski, A. Kolinski, and A. Godzik. 1997. Derivation and testing of pair potentials for protein folding. When is the quasichemical approximation correct? Protein Sci. **6**: 676–88.

142 Skolnick, J. and D. Kihara. 2001. Defrosting the frozen approximation: PROSPECTOR – a new approach to threading. Proteins **42**: 319–31.

143 Skolnick, J., D. Kihara, and Y. Zhang. 2004. Development and large scale benchmark testing of the PROSPECTOR_3 threading algorithm. Proteins **56**: 502–18.

144 Skolnick, J., A. Kolinski, and A. Ortiz. 2000. Derivation of protein-specific pair potentials based on weak sequence fragment similarity. Proteins **38**: 3–16.

145 Smith, T. F. and M. S. Waterman. 1981. Identification of common molecular subsequences. J. Mol. Biol. **147**: 195–7.

146 Söding, J. 2005. Protein homology detection by HMM-HMM comparison. Bioinformatics **21**: 951–60.

147 Sommer, I., A. Zien, N. von Öhsen, R. Zimmer, and T. Lengauer. 2002. Confidence measures for protein fold recognition. Bioinformatics **18**: 802–12.

148 Spolar, R. S. and M. T. Record. 1994. Coupling of local folding to site-specific binding of proteins to DNA. Science **263**: 777–84.

149 Sternberg, M. J., P. A. Bates, L. A. Kelley, and R. M. MacCallum. 1999. Progress in protein structure prediction: assessment of CASP3. Curr. Opin. Struct. Biol. **9**: 368–73.

150 Sunyaev, S., F. Eisenhaber, I. Rodchenkov, B. Eisenhaber, V. Tumanyan, and E. Kuznetsov. 1999. PSIC: profile extraction from sequence alignments with position-specific counts of independant observations. Protein Eng. **12**: 387–94.

151 Tanaka, H., M. Ishikawa, K. Asai, and A. Konagaya. 1993. Hidden Markov models and iterative aligners: study of their equivalence and possibilities. Proc. ISMB **1**: 395–401.

152 Tang, C. L., L. Xie, I. Y. Y. Koh, S. Posy, E. Alexov, and B. Honig. (2003). On the role of structural information in remote homology detection and sequence alignment: new methods using hybrid sequence profiles. J. Mol. Biol. **334**: 1043–62.

153 Tatusov, R., S. Altschul, and E. Koonin. 1994. Detection of conserved segments in proteins: Iterative acanning of sequence databases with alignment blocks. Proc. Natl Acad. Sci. USA **91**: 12 091–5.

154 Thiele, R., R. Zimmer, and T. Lengauer. 1999. Protein threading by recursive dynamic programming. J. Mol. Biol. **290**: 757–79.

155 Thompson, J. D., D. G. Higgins, and T. J. Gibson (1994a, Nov). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. **22**: 4673–80.

156 Thompson, J. D., D. G. Higgins, and T. J. Gibson. (1994b). Improved sensitivity of profile searches through the use of sequence weights and gap excision. Comput. Appl. Biosci. **10**: 19–29.

157 Tompa, P. (2003a). Intrinsically unstructured proteins. Trends Biochem. Sci. **27**: 527–533.

158 Tompa, P. (2003b). Intrinsically unstructured proteins evolve by repeat expansion. BioEssays **25**: 847–55.

159 Tosatto, S. C. E. 2005. The Victor/FRST function for model quality estimation. J. Comput. Biol. **12**: 1316–27.

160 Tosatto, S. C. E. and S. Toppo. 2006. Large-scale prediction of protein structure and function from sequence. Curr. Pharm. Des. **12**: 2067–86.

161 Tress, M., C.-H. Tai, G. Wang, I. Ezkurdia, G. López, A. Valencia, B. Lee, and R. L. Dunbrack. 2005. Domain definition and target classification for CASP6. Proteins **61 (Suppl. 7)**: 8–18.

162 van Holde, K. E., W. C. Johnson, and P. S. Ho. 1998. *Principles of Physical Biochemistry*. Prentice Hall, Upper Saddle River, NJ.

163 von Grotthuss, M., J. Pas, L. Wyrwicz, K. Ginalski, and L. Rychlewski. 2003. Application of 3D-Jury, GRDB, and Verify3D in fold recognition. Proteins **53 (Suppl. 6)**: 418–23.

164 von Öhsen, N. 2005. A novel profile-profile alignment method and its application in fully automated protein structure prediction. Ph. D. thesis, Ludwig Maximilians Universität München.

165 von Öhsen, N., I. Sommer, and R. Zimmer. 2003. Profile–profile alignment: a powerful tool for protein structure prediction. Pac. Symp. Biocomput. **8**: 252–63.

166 von Öhsen, N., I. Sommer, R. Zimmer, and T. Lengauer. 2004. Arby: automatic protein structure prediction using profile–profile alignment and

confidence measures. Bioinformatics **20**: 2228–35.

167 von Öhsen, N. and R. Zimmer. 2001. Improving profile-profile alignment via log average scoring. Proc. WABI **1**: 11–26.

168 Vucetic, S., C. J. Brown, A. K. Dunker, and Z. Obradovic. 2003. Flavors of protein disorder. Proteins **52**: 573–84.

169 Wallner, B. and A. Elofsson. 2003. Can correct protein models be identified? Protein Sci. **12**: 1073–86.

170 Wallner, B., H. Fang, T. Ohlson, J. Frey-Skött, and A. Elofsson. 2004. Using evolutionary information for the query and target improves fold recognition. Proteins **54**: 342–50.

171 Wang, G. and R. L. Dunbrack. 2004. Scoring profile-to-profile sequence alignments. Protein Sci. **13**: 1612–26.

172 Wang, G., Y. Jin, and R. L. Dunbrack. 2005. Assessment of fold recognition predictions in CASP6. Proteins **61 (Suppl. 7)**: 46–66.

173 Wang, Y., S. Bryant, R. Tatusov, and T. Tatusova. 2000. Links from genome proteins to known 3-D structures. Genome Res. **10**: 1643–7.

174 Ward, J. J., J. S. Sodhi, L. J. McGuffin, B. F. Buxton, and D. T. Jones. (2004). Prediction and functional analysis of native disorder in proteins from the three kingdoms of life. J. Mol. Biol. **337**: 635–45.

175 Waterman, M. and M. Vingron. 1994. Rapid and accurate estimates of statistical significance for sequence base searches. Proc. Natl Acad. Sci. USA **91**: 4625–8.

176 Webber, C. and G. J. Barton. 2001. Estimation of *P*-values for global alignments of protein sequences. Bioinformatics **17**: 1158–67.

177 Weston, J., C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble. (2005). Semi-supervised protein classification using cluster kernels. Bioinformatics **21**: 3241–7.

178 Wolf, Y., S. Brenner, P. Bash, and E. Koonin. 1999. Distribution of protein folds in the three superkingdoms of life. Genome Res. **9**: 17–26.

179 Wood, M. J. and J. D. Hirst. 2005. Protein secondary structure prediction with dihedral angles. Proteins **59**: 476–81.

180 Wu, T., C. Nevill-Manning, and D. Brutlag. 1999. Minimal-risk scoring matrices for sequence analysis. J. Comput. Biol. **6**: 219–35.

181 Xu, J. and M. Li. 2003. Assessment of RAPTOR's linear programming approach in CAFASP3. Proteins **53 (Suppl. 6)**: 579–84.

182 Xu, J., M. Li, D. Kim, and Y. Xu. 2003. Raptor: optimal protein threading by linear programming. J. Bioinform. Comput. Biol. **1**: 95–117.

183 Xu, J., M. Li, G. Lin, D. Kim, and Y. Xu. 2003. Protein threading by linear programming. Pac. Symp. Biocomput. **8**: 264–75.

184 Yang, A. and B. Honig. 2000. An integrated approach to the analysis and modeling of protein sequences and structures. III. A comparative study of sequence conservation in protein structural families using multiple structural alignments. J. Mol. Biol. **301**: 691–711.

185 Yona, G. and M. Levitt. 2002. Within the twilight zone: a sensitive profile–profile comparison tool based on information theory. J. Mol. Biol. **315**: 1257–75.

186 Zemla, A. 2003. LGA: A method for finding 3D similarities in protein structures. Nucleic Acids Res. **31**: 3370–4.

187 Zhang, L. and J. Skolnick. 1998. How do potentials derived from structural databases relate to "true" potentials? Protein Sci. **7**: 112–22.

188 Zhang, Y., A. K. Arakaki, and J. Skolnick. 2005. TASSER: An automated method for the prediction of protein tertiary structures in CASP6. Proteins **61 (Suppl. 7)**: 91–8.

189 Zhang, Y. and J. Skolnick. (2004). Automated structure prediction of weakly homologous proteins on a genomic scale. Proc. Natl Acad. Sci. USA **101**: 7594–9.

190 Zhang, Y. and J. Skolnick. (2004b). Scoring function for automated assessment of protein structure template quality. Proteins **57**: 702–10.

191 Zhou, H. and Y. Zhou. 2005. Fold recognition by combining sequence profiles derived >from evolution and from depth-dependent structural alignment of fragments. Proteins **58**: 321–8.

12
De Novo Structure Prediction: Methods and Applications

Richard Bonneau

1 Introduction

1.1 Scope of this Review and Definition of *De Novo* Structure Prediction

This review will focus on the questions: (i) what are the features common to methods that represent the current state of the art in *de novo* structure prediction and (ii) how can these methods benefit biologists whose primary aim is a systems-wide description of a given organism or system of organisms. The role and capabilities of *de novo* structure prediction as well as the relationship of *de novo* structure prediction to other sequence and structure-based methods is far from simple. The literature on this subject is rapidly evolving; for balance in coverage and opinion the reader is also referred to recent reviews of *de novo* structure prediction methods [11,27,32,39,50].

Many methods that are today referred to as *de novo* have alternately or previously been referred to *ab initio* or "new folds" methods. For the purpose of this review I will classify a method as *de novo* structure prediction if that method does not rely on homology between the query sequence and a sequence in the Protein Data Bank (PDB) to create a template for structure prediction. *De novo* methods, by this definition, are forced to consider much larger conformational landscapes than fold recognition and comparative modeling techniques that limit the exploration of conformational space to those regions close to the initial structural template or templates.

Another common pedagogical distinction between structure prediction methods has been the distinction between methods based on statistical principles, on the one hand, and physical or first principles, on the other hand. I will not discuss this distinction here at great length except for noting that one of the shortcomings of this artificial division is that most effective structure prediction methodologies are in fact a combination of these two camps. For example, several methods that are described as based on physical or first-principles employ energy functions and parameters that are statistical approximations of data (e.g. the Lennard–Jones representation of van der

Bioinformatics - From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

Waals forces is often thought of as a physical potential, but is a heuristic fit to data). Most current successful *de novo* structure prediction methods fall into the statistics camp. A more useful distinction may be the distinction between reduced complexity models and models that use atomic detail. Throughout this chapter I will discuss low-resolution (models containing drastic reductions in complexity such as unified atoms and centroid representations of side-chain atoms) and high-resolution methods (methods that represent protein and sometimes solvent in full atomic detail) focusing on this practical classification/division of methods in favor of distinctions based on a given method's derivation or parameterization.

1.2 The Role of Structure Prediction in Biology

What is the main application of structure prediction to biology? At present this is an open question that will take many years to develop, as the answer relies on the relative rate of progress in several fields. In short, I will argue that the main current application of structure prediction in biology lies in understanding protein function. Structure predictions can offer meaningful biological insights at several functional levels depending on the method used to generate the structure prediction, the expected resolution and the comprehensiveness or scale on which predictions are available for a given system.

At the highest levels of detail/accuracy (comparative modeling) there are several similarities between the uses of experimental and computational/predicted protein structure and the types of functional information that can be extracted from models generated by both methods [4]. For example, experimentally determined structures and structures resulting from comparative modeling can be used to help understand the details of protein function at an atomic scale, map conservation and mutagenesis data onto a structural framework, and explore detailed functional relationships between protein with similar folds or active sites.

At the other end of the prediction resolution spectrum, *de novo* structure prediction and fold recognition methods produce models of lower resolution than comparative models (see Chapter 10). These models can be used to assign putative functions to proteins for which little is known [15]. At the most basic level we can use structural similarities between a predicted structure and known structures to explore possible distant evolutionary relationships between query proteins of unknown function and other well-studied proteins for which structures have been experimentally determined. A query protein is likely to share some functional aspects with proteins in the PDB that show strong structure–structure matches to a high confidence predicted structure for that protein. This is based on the assumption that detectable structure relationships are conserved across a greater evolutionary distance than are detectable sequence similarities. This assumption is well supported by multiple surveys of the distributions of folds and their related functions in the PDB [48, 68, 76, 83]. The relationship between fold and function, however, is by no means a simple subject, and I refer the reader to several works that discuss this relationship in greater detail [56, 70, 84, 107]. Another way of exploring the functional significance of high confidence predicted structures is to use libraries of three-dimensional (3-D) functional motifs to search for conserved active site or functional motifs on the predicted structures [33, 72, 103]. Both basic methods, fold–fold matching and the use of small 3-D functional motif searches, can in principle be combined to form the basis for deriving functional hypothesis from predicted structure, thereby extending the completeness of genome annotations based only on primary sequence. For more details on how to infer protein function from protein structure, see Chapter 34.

1.3 *De novo* Structure Prediction in a Genome Annotation Context, Synergy with Other Methods

To date, the annotation of protein function in newly sequenced genomes relies on a large array of tools based ultimately on primary sequence analysis [3, 9, 19, 100]. These tools have afforded great progress in genome annotation including large improvements in gene detection, sequence alignment and detection of homologous sequences across genomes as well as the creation of databases of common protein families and primary sequence functional motifs. Comparative modeling methods have been highly successful on many fronts, creating large databases of highly accurate structure predictions for many organisms, but are based on primary sequence matches between PDB and query sequences [87] (see Chapter 10). Primary sequence methods also exist for the prediction of basic local structure qualities (some of these patterns being lower complexity patterns) of sequences such as the location of coiled-coil, transmembrane and disordered regions [52, 80, 99, 104]. Efforts to use *de novo* structure prediction (and/or fold recognition) must employ these sequence-based methods, as these methods provide a solid foundation on which all *de novo* methods discussed herein are reliant (see Figure 1). Any organization of these methods into an annotation pipeline must properly account for the fact that the accuracy/reliability is quite different between sequence and structure-based methods. One approach is to use structure prediction as part of a hierarchy where methods yielding high-confidence results are exhausted prior to computationally expensive and less accurate *de novo* structure prediction and fold recognition [12] . I will describe some early results from these approaches/pipelines that include structure prediction, the

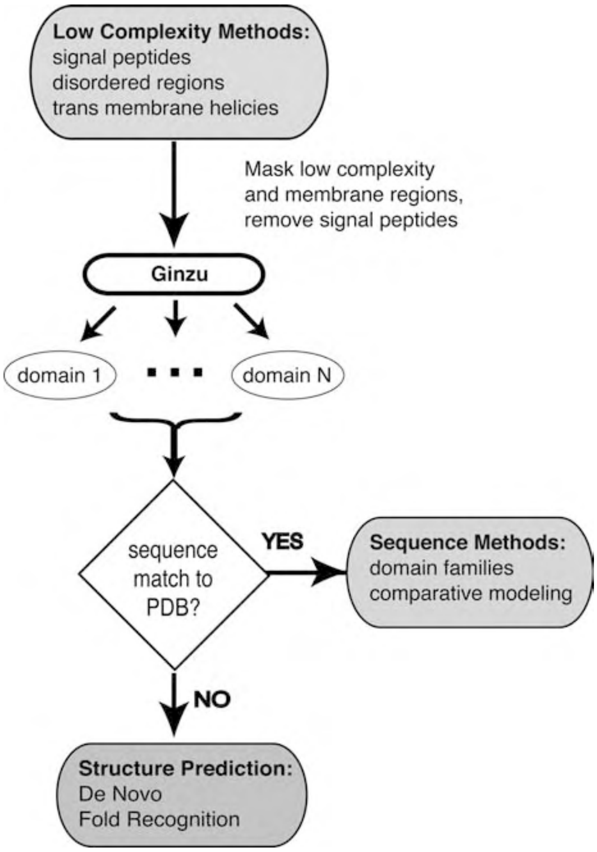


Figure 1 Idealized proteome structure annotation pipeline. Low-complexity regions such as transmembrane helices, signal peptides and disordered regions are masked, and domains dominated by these low-complexity or transmembrane sequence are treated separately. Remaining sequences are parsed to separate regions into structural domains to the degree that such domains are detectable (here, Ginzu is shown as the domain parsing algorithm, see Figure 4). Domains that do not have strong sequence matches to the PDB or other matches to well-annotated domains (Pfam, COG) are forwarded to structure-based methods. The use of structure prediction methods is positioned within this hierarchy of methods to increase comprehension of the resulting annotation without compromising the results obtained by sequence-based methods.

details of these pipelines and the technical and research challenges that remain in applying these pipelines to genome annotation [6,45,86].

The need for methods for predicting transmembrane proteins and understanding membrane–protein interactions is not discussed in this work (see Chapter 9 for this topic), the focus here is instead on soluble domains (including soluble domains excised from proteins containing transmembrane regions). Part of the difficulty in predicting transmembrane protein structure lies in the paucity of membrane protein structures deposited in the PDB

[28, 99]. It is only with access to the PDB, an ideal and comprehensive gold standard, by many criteria, that we can approach the problem of predicting soluble protein structure.

2 Core Features of Current Methods of *De Novo* Structure Prediction

We will now discuss core concepts that are common to multiple successful current *de novo* methods. This review is not intended to be encyclopedic and will invariably fail to mention several methods that are innovative and/or accurate in its attempt to focus on core concepts instead of distinct methodologies. The omission of any specific method should not be interpreted as commentary on the relative accuracy of the omitted method, but is simply due to the scope of this work and the state of rapid development in this field.

2.1 Rosetta *De Novo*

Throughout this work I will use examples of key concepts in *de novo* structure prediction with several examples drawn from the Rosetta *de novo* structure prediction protocol and will thus provide a brief overview of Rosetta before continuing to discuss key elements of the procedure in greater detail [13, 90, 97, 98] (see Figures 2 and 3). Results from the fourth and fifth Critical Assessments of Structure Prediction (CASP4, CASP5 and CASP6; see also Chapter 11) have shown that Rosetta is currently one of the best methods for *de novo* protein structure prediction and distant fold recognition [16, 18, 26, 65]. Rosetta was initially developed as a computer program for *de novo* fold prediction, but has been expanded to include design, docking, experimental determination of structure from partial datasets, protein–protein interaction and protein–DNA interaction prediction [25, 41, 42, 57, 59, 60, 88, 89]. When referring to Rosetta in this work I will be primarily referring to the *de novo* or *ab initio* mode of the Rosetta code base. Early progress in high-resolution structure prediction has been achieved via combinations of low-resolution approaches (for initially searching the conformational landscape) and higher-resolution potentials (where atomic detail and physically derived energy functions are employed). Thus, Rosetta structure prediction is carried out in two phases: (i) a low-resolution phase where overall topology is searched using a statistical scoring function and fragment assembly, and (ii) an atomic-detail refinement phase using rotamers and small backbone angle moves, and a more physically relevant (detailed) scoring function. The algorithms for searching the landscape are Monte-Carlo-type in both phases.

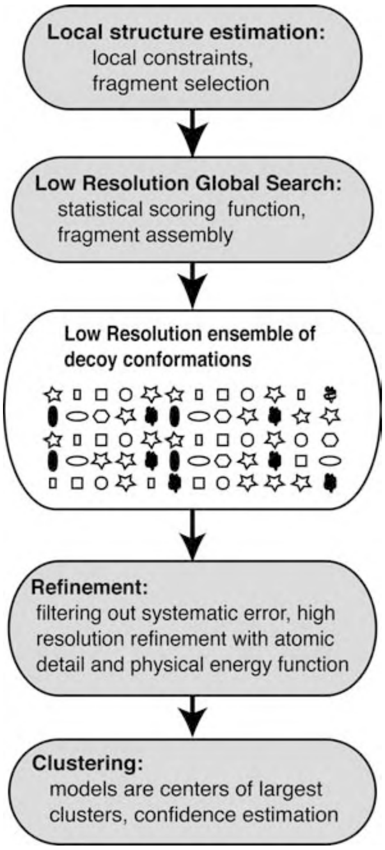

Local structure estimation:
local constraints,
fragment selection
Low Resolution Global Search:
statistical scoring function,
fragment assembly
Low Resolution ensemble of
decoy conformations
Refinement:
filtering out systematic error, high
resolution refinement with atomic
detail and physical energy function
Clustering:
models are centers of largest
clusters, confidence estimation

Figure 2 Schematic outline of Rosetta structure prediction protocol. Single sequences enter at the top of this schematic and confidence-ranked structure predictions are produced by the last/bottom step.

In the first phase, Rosetta *de novo* (Rosetta) uses information from the PDB to estimate the possible conformations for local sequence segments. The procedure first generates libraries of local sequence fragments excised from the PDB on the basis of local sequence similarity (three- and nine-residue matches between the query sequence and a given structure in the PDB). See Figure 1 for a schematic overview of the low-resolution (or fold prediction) phase of the Rosetta method, and see Tables 1 and 2 for a complete description of the Rosetta score. Rosetta fragment generation works well even for sequences that have no homologs in the known sequence databases; the structures in the PDB cover possible local sequence well at the three- and nine-residue length according to the current method. Rosetta then assembles these pre-computed local structure fragments by minimizing a global scoring function that favors hydrophobic burial/packing, strand pairing, compactness and highly proba-

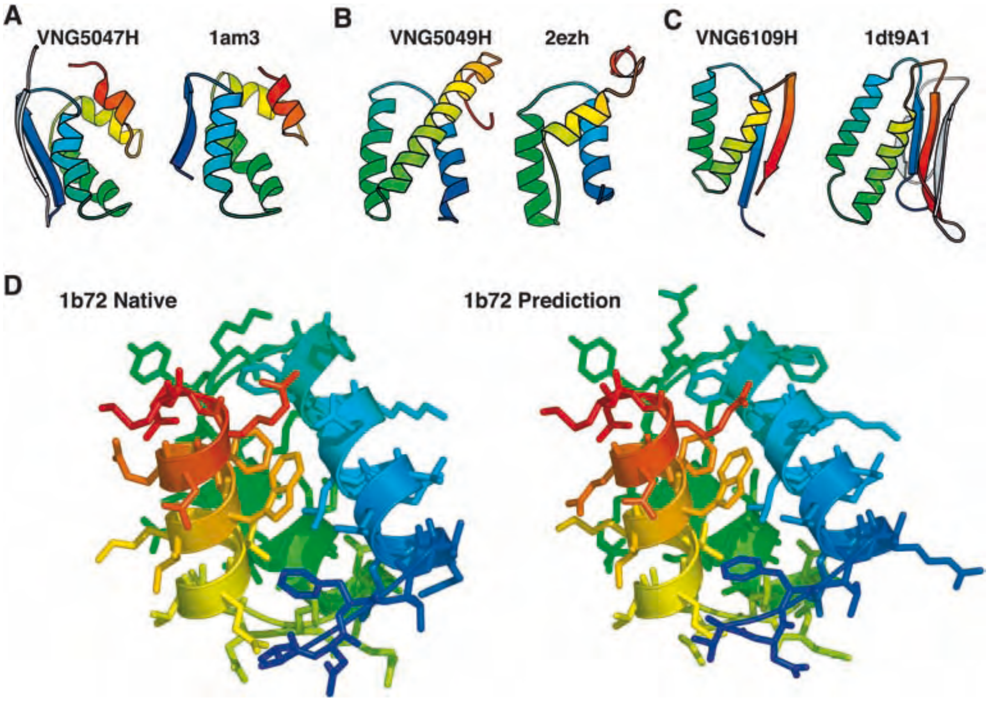


Figure 3 Examples of *de novo* structure predictions generated using Rosetta. (A–C) Examples from our genome-wide prediction of domains of unknown function in *Halobacterium* NRC-1 [12]. In each case the predicted structure is shown next to the correct native. For (A–C) only the backbone ribbons are shown, as these predictions were not refined using the all-atom potential and are examples of the utility of low-resolution prediction in determining function. (D) A recent prediction where high-resolution refinement subsequent to the low-resolution search produced the lowest energy conformation, a prediction of unprecedented accuracy (provided by Phil Bradley) [17].

ble residue pairings. The Rosetta score for this initial low-resolution stage is described in its entirety in Table 1. For the second, refinement, stage centroid representations of amino acid side-chains are replaced with atomic detail (rotamer representations). The scoring function used during this refinement phase includes solvation terms, hydrogen bond terms and other terms with direct physical interpretation. See Table 2 for a full description of the all-atom Rosetta score. Features of the high- and low-resolution phases of the Rosetta method are described below as I discuss key components of *de novo* structure prediction universal to all successful methods.

Using Rosetta generated structure predictions we were able to recapitulate many functional insights not evident from sequence based methods alone [14, 15]. We have reported success in annotating proteins and protein families without links to known structure with Rosetta [8, 14]. Various aspects of this

Table 1 Low-resolution, centroid-based Rosetta scoring function[a]

Name	Description (physical origin)	Functional form	Parameters (values)
env[b]	residue environment (solvation)	$\sum_i - \ln[P(aa_i \vert nb_i)]$	i = residue index aa = amino acid type nb = number of neighboring residues[c] (0, 1, 2, ..., 30. >30)
pair[b]	residue pair interactions (electrostatics, disulfides)	$\sum_i \sum_{j>i} - \ln \left\lfloor \frac{P(aa_i, aa_j \vert s_{ij} d_{ij})}{P(aa_i \vert s_{ij} d_{ij}) P(aa_j \vert s_{ij} d_{ij})} \right\rfloor$	i, j = residue indices aa = amino acid type d = centroid–centroid distance (10–12, 7.5–10, 5–7.5, <5 Å) s = sequence separation (>8 residues)
vdw[g]	steric repulsion	$\sum_i \sum_{j>i} \frac{(r_{ij}^2 - d_{ij}^2)^2}{r_{ij}^2}; \quad d_{ij} < r_{ij}$	i, j = residue (or centroid) indices d = interatomic distance r = summed van der Waals radii[h]
rg	radius of gyration (van der Waals attraction; solvation)	$\sqrt{\langle d_{ij}^2 \rangle}$	i, j = residue indices d = distance between residue centroids
cbeta	C_β density (solvation; correction for excluded volume effect introduced by simulation)	$\sum_i \sum_{sh} - \ln \left\lfloor \frac{P_{\text{compact}}(nb_{i,sh})}{P_{\text{random}}(nb_{i,sh})} \right\rfloor$	i = residue index sh = shell radius (6, 12 Å) nb = number of neighboring residues within shell[f] $P_{compact}$ = probability in compact structures assembled from fragments P_{random} = probability in structures assembled randomly from fragments

overall protocol will be reviewed in greater detail below. We also encourage the reader to refer to several prior works where the Rosetta method is described in its entirety.

2.2 Evaluation of Structure Predictions

In general the most effective methods for predicting structure *de novo* depend on parameters ultimately derived from the PDB. Several methods use the PDB directly to estimate local sequence and even explicitly use fragments of local sequence from the PDB to build global conformations. These uses of the PDB require that methods be tested using structures not present in the sets

Table 1 continued

Name	Description (physical origin)	Functional form	Parameters (values)
SS[d]	strand pairing (hydrogen bonding)	Scheme A: $\mathrm{SS}_{\phi,\theta}$ + SS_{hb} + SS_{d} Scheme B: $\mathrm{SS}_{-\phi,\theta}$ + SS_{hb} + $\mathrm{SS}_{d\sigma}$ where $SS_{\phi,\theta} = \sum_{m} \sum_{n>m}$ $- \ln[P(\phi_{mn}, \theta_{mn} \vert d_{mn}, sp_{mn}, s_{mn})]$ $SS_{hb} = \sum_{m} \sum_{n>m}$ $- \ln[P(hb_{mn}, \vert d_{mn}, s_{mn})]$ $SS_{d} = \sum_{m} \sum_{n>m}$ $- \ln[P(d_{mn}, \vert s_{mn})]$ $SS_{d\sigma} = \sum_{m} \sum_{n>m}$ $- \ln[P(d_{mn}, \sigma_{mn} \vert \rho_{m}, \rho_{n})]$	m, n = strand dimer indices; dimer is two consecutive strand residues $\hat{V}$ = vector between first N and last C atom of dimer $\hat{m}$ = unit vector between $\hat{V}m$ and $\hat{V}n$ midpoints $\hat{x}$ = unit vector along carbon-oxygen bond of first dimer residue $\hat{y}$ = unit vector along oxygen-carbon bond of second dimer residue ϕ, θ = polar angles between $\hat{V}m$ and $\hat{V}n$ (10, 36° bins) $hb = dimertwist,$ $\sum_{k=m,n} 0.5(\vert\hat{m} \cdot \hat{x}_k\vert + \vert\hat{m} \cdot \hat{y}_k\vert)$ (<0.33, 0.33–0.66, 0.66–1.0, 1.0–1.33, 1.33–1.6, 1.6–1.8, 1.8–2.0) d = distance between $\hat{V}m$ and $\hat{V}n$ midpoints (<6.5 Å) σ = angle between $\hat{V}m$ and $\hat{M}$ (18° bins) sp = sequence separation between dimer-containing strands (<2, 2–10, >10 residues) s = sequence separation between dimers (>5 or >10) ρ = mean angle between vectors $\hat{m}$, $\hat{x}$ and $\hat{m}, \hat{y}$ (180° bins)
sheet[e]	strand pair arrangement into sheets	$- \ln[P(n_{\mathrm{sheets}} n_{\mathrm{lone\_strands}} \vert n_{\mathrm{strands}})]$	n_{sheets} = number of sheets $n_{\mathrm{lone\_strands}}$ = number of unpaired strands n_{strands} = total number of strands
HS	helix-strand packing	$\sum_{n} \sum_{n} - \ln[P(\phi_{mn}, \psi_{mn} \vert sp_{mn} d_{mn})]$	m = strand dimer index; dimer is two consecutive strand residues n = helix dimer index; dimer is central two residues of four consecutive helical residues $\hat{V}$ = vector between first N and last C atom of dimer ϕ, θ = polar angles between $\hat{V}m$ and $\hat{V}n$ (36° bins) sp = sequence separation between dimer-containing helix and strand (binned <2, 2–10, >10 residues) d = distance between $\hat{V}m$ and $\hat{V}n$ midpoints (<12 Å)

of protein structures used to train these methods (or present in the sets of structures used to predict local structure fragments). The first such evaluation of structure prediction, CASP (see Chapter 11 for a more detailed description), showed that published estimates of prediction error were smaller than prediction error measured on a set of novel proteins outside the training set (this is not surprising given the difficulties of avoiding overfitting in as complex a data space as protein structure) [64]. Indeed, early experiments showed that no methods for *de novo* structure prediction were effective outside of carefully chosen benchmarks containing only the smallest proteins. Spurred on by these early evaluations the field returned to the drawing board and two years later produced multiple methods with much higher accuracies in the new folds or *de novo* category (CASP3) [73,75,82]. Thus, the CASP experiments proved to be invaluable to the field at that point in the development of the field, provoking a renewed interest in the *de novo* structure prediction and properly realigned interest in techniques according to effectiveness.

Arguably, CASP has the flaw that predictors are allowed to intervene and manually curate their predictions prior to submission to the CASP evaluators. Thus, the results of CASP are a convolution of: (i) the art of prediction (each group's intuition and skill using their tools) and (ii) the relative performance

Footnotes to Table 1:

[a] The individual components in the Rosetta score (the score used by Rosetta during low-resolution/centroid mode *de novo* structure prediction) are given as described originally in Simons [96–98].

[b] Binned function values are linearly interpolated, yielding analytic derivatives.

[c] Neighbors within a 10 Å radius. Residue position defined by C_β coordinates (C_α for glycine).

[d] Interactions between dimers within the same strand are neglected. Favorable interactions are limited to preserve pairwise strand interactions, i.e. dimer m can interact favorably with dimers from at most one strand on each side, with the most favorable dimer interaction ($SS_{\phi s\theta}$ + SS_{hb} + SS_{d}) determining the identity of the interacting strand. $SS_{d\sigma}$ is exempt from the requirement of pairwise strand interactions. SS_{hb} is evaluated only for m, n pairs for which $SS_{\phi,\theta}$ is favorable. $SS_{d\sigma}$ is evaluated only for m, n pairs for which $SS_{\phi,\theta}$g and SS_{hb} are favorable. A bonus is awarded for each favorable dimer interaction for which $|m \; - - n| > 11$ and strand separation is more than eight residues

[e] A sheet is comprised of all strands with dimer pairs less than 5.5 Å apart, allowing each strand having at most one neighboring strand on each side. Discrimination between alternate strand pairings is determined according the most favorable dimer interaction. Probability distributions fitted to $c(n_{\text{strands}}) - 0.9n_{\text{sheets}} - 2.7n_{\text{lone\_strands}}$ where $c(n_{\text{strands}}) = (0.07, 0.41, 0.43, 0.60, 0.61, 0.85, 0.86, 1.12)$.

[f] Residue position defined by C_β coordinates (C_α for glycine).

[g] Not evaluated for atom (centroid) pairs whose interatomic distance depends on the torsion angles of a single residue.

[h] Radii determined from (i) 25th closest distance seen for atom pair in pdbselect25 structures, (ii) the fifth closest distance observed in X-ray structures with better than 1.3-Å resolution and less than 40% sequence identity or (iii) X-ray structures of less than 2 Å resolution, excluding i, i + 1 contacts (centroid radii only).

of the core methods (the performance of each method in an automatic setting). Although this convolution reflects the reality when workers aim to predict proteins of high interest, such as proteins involved in a specific function or proteins critical to a given disease or process being experimentally studied, it does not reflect the demands placed on a method when trying to predict whole genomes, where the shear number of predictions does not allow for much manual intervention. Several additional tests similar to CASP (in that they are blind tests of structure prediction) have been organized in response to the concerns of many that it is important to remove the human aspects of CASP. The Critical Assessment of Fully Automatic Structure Prediction (CAFASP) is an experiment running parallel with CASP that aims to test fully automated methods' performance on CASP targets, mainly testing servers instead of groups [35, 36]. Several groups have also raised concerns that there are problems associated with the small numbers of proteins tested in each CASP experiment, and thus EVA and LiveBench were organized to test methods using larger numbers of proteins [20, 92, 94]. Both use proteins that have structures that are unknown to the participating prediction groups, but that have been recently submitted to the PDB and are not open to the public at the time their sequences are released to those participating in LiveBench or EVA. The participating groups then have the time it takes for the new PDB entries to be validated to predict the structures. Although groups with amazing computer-hacking skills could in principle access this information, these efforts effectively create a CAFASP equivalent for a larger number of proteins.

All four of these tests of prediction methods, as well as benchmarks carried out by authors of any methods in question, are valuable ways of judging the performance of *de novo* methods. The methods, and elements of methods, I describe herein are generally accepted to be the best performers by the five above measures (four blind tests and author benchmarks). I will not focus on the details of the CASP, CAFASP, EVA and LiveBench methods, as they are described in detail elsewhere and instead attempt to focus on common elements of top performing methods.

2.3 Domain Prediction is Key

As the size of a protein increases, so to does the size of the conformational space associated with that protein. Thus, *de novo* methods, which must sample this space, have run times that increase dramatically with sequence length. Current *de novo* methods are limited to proteins and protein domains less than 150 amino acids in length (with Rosetta the limit is around 150 residues for α/β proteins, 80 for β-folds and more than 150 residues for α-only-folds). This limit means that roughly half of the protein domains seen so far in the

Table 2 All-atom Rosetta scoring function: the components of the all-atom score (centroids are expanded using a rotamer description of side-chains) [31, 44, 58, 62, 77, 105]

Name	Description	Functional form	Parameters, variables	References
rama	Ramachandran torsion preferences	$\sum_i -\ln[P(\phi_i, \psi_i \vert aa_i ss_i)]$	i = residue index ϕ, ψ = backbone torsion angles (10°, 36° bins) aa = amino acid type ss = secondary structure type[a]	Bowers, 2000 [16a]
LJ[c]	Lennard–Jones interactions	$\sum_i \sum_{j>i} \begin{cases} \left[\left(\frac{r_{ij}}{d_{ij}}\right)^{12} - 2\left(\frac{r_{ij}}{d_{ij}}\right)^{6}\right] e_{ij} \\ \text{if } \frac{d_{ij}}{r_{ij}} > 0.6 \\ \left[-8759.2\left(\frac{d_{ij}}{r_{ij}}\right) + 5672.0\right] e_{ij}, \\ \text{else} \end{cases}$	i,j = residue indices d = interatomic distance e = geometric mean of atom well depths[d] r = summed van der Waals radii[e]	Kuhlman, 2000 [59]
hb[f]	hydrogen bonding	$\sum_i \sum_j (-\ln[p(d_{ij} \vert h_j ss_{ij})]$ $-\ln[P(\cos\theta_{ij} \vert d_{ij} h_j ss_{ij})]$ $-\ln[P(\cos\theta_{ij} \vert d_{ij} h_j ss_{ij})]$ $-\ln[P(\cos\psi_{ij} \vert d_{ij} h_j ss_{ij})]$	i = donor residue index j = acceptor residue index d = acceptor-proton interatomic distance h = hybridization (sp^2, sp^3) ss = secondary structure type[g] θ = proton–acceptor–acceptor base bond angle ψ = donor–proton–acceptor bond angle	Kortemme, 2003 [58]
solv	solvation	$\sum_i \left[\Delta G_i^{\text{ref}} - \sum_j \left(\frac{2\Delta G_i^{\text{free}}}{4\pi^{3/2}\lambda_i r_{ij}^2} e^{-d_{ij}^2} V_j\right.\right.$ $\left.\left. + \frac{2\Delta G_i^{\text{freee}}}{4\pi^{3/2}\lambda_j r_{ij}^2} e^{-d_{ij}^2} V_i\right)\right]$	i, j = atom indices d = distance between atoms r = summed van der Waals radii[e] λ = correlation length[h] V = atomic volume[h] ΔG^{ref}, ΔG^{free} = energy of a fully solvated atom[h]	Lazaridis, 1999 [62]
pair	residue pair interactions (electrostatics, disulfides)	$\sum_i \sum_{j>i} -\ln\left[\frac{P(aa_i, aa_j) \vert d_{ij})}{P(aa_i \vert d_{ij}) P(aa_i \vert d_{ij})}\right]$	i,j = residue indices aa = amino acid type d = distance between residues[i]	Kuhlman, 2000 [59]
dun	rotamer self energy	$\sum_i -\ln\left[\frac{P(\text{rot}_i \vert \phi_i \psi_i) P(aa_i \vert \phi_i, \psi_i)}{P(aa_i)}\right]$	i,j = residue indices rot = Dunbrack backbone-dependent rotamer aa = amino acid type ϕ, ψ = backbone torsion angles	Dunbrack, 1997 [31]
ref	unfolded state reference energy	$\sum_{aa} n_{aa}$	aa = amino acid type n = number of residues	Kuhlman, 2000 [59]

PDB are within the size limit of *de novo* structure prediction. Two approaches to circumventing this size limitation are: (i) increasing the size range of *de novo* structure prediction and (ii) dividing proteins into domains prior to attempting to predict structure. Dividing query sequences into their smallest component domains prior to folding is one straightforward way to dramatically increase the reach of *de novo* structure prediction. For many proteins domain divisions can be easily found (as would be the case for a protein where one domain was unknown and one domain was a member of a well-known protein family) while several domains remain beyond our ability to correctly detect them. The determination of domain family membership and domain boundaries for multi-domain proteins is a vital first step in annotating proteins on the basis of primary sequence and has ramifications for several aspects of protein sequence annotation; multiple works describe methods for detecting such boundaries. In short, most protein domain parsing methods rely on hierarchically searching for domains in a query sequence with a collection of primary sequence methods, domain library searches and matches to structural domains in the PDB [26,55,66].

Some notable works use coarse-grained structural simulations/predictions coupled with methods for assigning structural domain boundaries to 3-D structures to detect protein domains from sequence. The guiding principle behind this approach is that very low-resolution predictions will pick up overall patterns of the polypeptide packing into distinct structural domains. Another recent work attempted to use local sequence signals to detect structure domain boundaries under the assumption that there would be detectable differences in local sequence propensities at domain boundaries [37]. As of yet these

Footnotes to Table 2:

[a] All binned function values are linearly interpolated, yielding analytic derivatives, except as noted.

[b] Three-state secondary structure type as assigned by DSSP.

[c] Not evaluated for atom pairs whose interatomic distance depends on the torsion angles of a single residue.

[d] Well depths taken from CHARMm19 parameter set (Neria 1996 [77]).

[e] Radii determined from fitting atom distances in protein X-ray structures to the 6–12 Lennard–Jones potential using CHARMm19 well depths.

[f] Evaluated only for donor acceptor pairs for which $1.4 \leq d \quad \leq 3.0$ and $90^\circ \leq \psi, \theta \quad \leq 180^\circ$. Side-chain hydrogen bonds in involving atoms forming main-chain hydrogen bonds are not evaluated. Individual probability distributions are fitted to eighth-order probability distributions and analytically differentiated.

[g] Secondary structure types for hydrogen bonds are assigned as helical ($j \; - - \; i = 4$, main-chain), strand: ($|j \; - - \; i| \quad > 4$, main-chain) or other.

[h] Values taken from Lazaridis and Karplus [62].

[i] Residue position defined by C_β coordinates (C_α of glycine).

* Also described in Rohl 2005 [90].

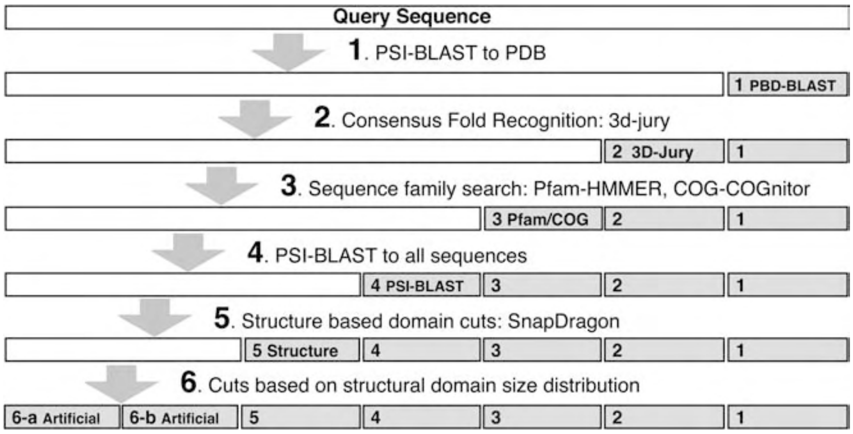

Query Sequence
1. PSI-BLAST to PDB
1 PBD-BLAST
2. Consensus Fold Recognition: 3d-jury
2 3D-Jury
1
3. Sequence family search: Pfam-HMMER, COG-COGnitor
3 Pfam/COG
2
1
4. PSI-BLAST to all sequences
4 PSI-BLAST
3
2
1
5. Structure based domain cuts: SnapDragon
5 Structure
4
3
2
1
6. Cuts based on structural domain size distribution
6-a Artificial
6-b Artificial
5
4
3
2
1

Figure 4 Schematic outline of an ideal hierarchical approach to domain parsing. Methods with higher reliability are used first, with sequence matches to the PDB being the highest-quality information. As higher-reliability/interpretability methods are exhausted, noisier methods are used (such as parsing multiple sequence alignments, step 4, and guessing domain boundaries based on the distribution of domain sizes in the PDB). Sequence regions hit by higher confidence methods (represented as gray rectangles) are masked and the remaining sequence (represented by white rectangles) are forwarded onto the remaining methods. Steps 1–4 and 6 are currently implemented in the Ginzu program; step 5 (adding sequence homolog independent methods such as structure-based domain parsing from sequence to the procedure) represent future work. Although we recognize domains in this schematic from left to right this direction is merely schematic, and Ginzu recognizes and parses domains in a fully general (discontinuous, depending on where the strong hits are at any given level) manner.

methods have unacceptably high error rates and are far too computationally demanding for use in genome wide predictions (David Kim, personal communication) [38]. In spite of the limitations mentioned above, these methods (that are not dependent on detecting sequence homologs for a given query sequence) are attractive for proteins that have no detectable homologs or matches to protein domain families and future work on this front could increase the number of proteins within reach of *de novo* methods considerably. It is likely that a method which successfully combines these coarse structure-based methods with existing sequence-based methods into a hierarchically organized domain detection program (e.g. Ginzu) will eventually outperform any existing method at domain parsing and greatly increase the accuracy of downstream structure prediction. Figure 4 shows a schematic domain detection program (this schematic is implemented as the program Ginzu).

2.4 Local Structure Prediction and Reduced Complexity Models are Central to Current *De Novo* Methods

Several methods for reducing the combinatorial complexity of the protein folding problem have been employed including lattice models (confining possible special coordinates to a predefined 3-D grid) and several discrete-state off-lattice models (e.g. reducing degrees of freedom along the backbone to a set of discrete angles). For a more exhaustive description of these methods and their reduced-complexity move sets I refer the reader to earlier reviews of *de novo* structure prediction methods [11, 27].

Instead, I will focus on the use of local structure information to constrain global structure prediction simulations to only conformations consistent with local structure prediction. Local sequences excised from protein structures often have stable structures in the absence of their global contacts, demonstrating that local sequences can have a strong, sequence-dependent, structural bias towards one or more well defined structures [10, 24, 69, 74, 106]. This experimental observation is a result of the fact that the polypeptide chain is heavily constrained by local structure bias in a sequence dependent manner. The strength of this local, sequence-dependent, structure bias can vary from strong (a local sequence that exhibits a single well defined local structure) to weak (local sequences that are disordered or completely determined by their global environment) with most protein sequences falling into some intermediate regime (local sequences that fold into multiple well-defined local structures depending on their global environment) [21, 46]. Prediction methods that accurately predict the type, strength and possible multiplicity of local structure bias for any given query sequence segment drastically reduce the size of the available conformational landscape. Using either fragment substitution (assembling fragments of local structure) as a move set or local structure constraints derived from predicted local structure also has the advantage that the subsequent global search is limited to protein-like regions of the conformational landscape (helices, correct chirality of secondary strand packing, strands and sheets with correct twist, etc.).

There are two main ways to use local structure prediction as an overriding/hard constraint on the global search: (i) using fragments to build up global structures (local structure defining the moveset) and (ii) using local structure as a hard constraint (local structure heavily modifying the objective function).

Rosetta explicitly uses fragments of three and nine residues of local structure to build global structures via fragment assembly. Prior to a Rosetta simulation a library of local structure fragments is generated such that several fragments (25–200) of different local structure are pre-computed for every possible three- and nine-residue window along the query. The simulation (the search for low-energy conformations given the Rosetta scoring function) consists primarily of randomly selecting three- and nine-residue windows along the query and replacing torsion angles at that three- or nine-residue window with torsion angles taken from a different fragment for that position. These fragments are pulled from a nonredundant version of the PDB on the basis of local sequence similarity to the query sequence [97]. This work was inspired by careful studies of the relationship between local sequence and local structure [46], that demonstrated that this relationship was highly variable on a sequence-specific basis and that there is a great deal of sequence-specific local structure that could be recognized even in the absence of global homology. The selection of fragments of local structure on the basis of local sequence matches dramatically reduces the size of the accessible conformational landscape. In practice we see that, as desired, for some local sequence segments there is a strong bias towards a single local structure in the computed local structure fragments, while other local sequences exhibit a wide range of local conformations in the fragment library. Using fragment substitution as a moveset to optimize Rosetta's objective function has one major drawback: as the structure collapses (forms many contacts favorable according to the energy function) late in the simulation the acceptance rate of fragment moves becomes unworkably small. This is due to the fact that the substitution of six or 18 backbone dihedral angles creates large perturbations to the Cartesian coordinates of parts of the protein distant along the polypeptide chain. The likelihood that such large perturbations cause steric clashes and break energetically favorable contacts late in a given simulation is exceedingly large. To recover effective minimization of the Rosetta score after initial collapse several additional move types have been added to the Rosetta moveset. The simplest move type consists of small angle moves (within populated regions of the Ramachandran map). Additional moves, descriptively named "chuck", "wobble" and "gunn" moves, aim to perform fragment insertions that have small effects far from the insertion. These additional move types are also critical to the modeling of loops in homology modeling and are described in detail elsewhere [89].

The TASSER method smoothly combines fragments of aligned protein structure (from threading runs) with regions of unaligned proteins (represented on a lattice for computational efficiency) to effectively scale between the fold recognition and *de novo* regime [108]. Other notable uses of local structure fragments include the use of I-sites to select fragments that are then fed to Rosetta as described by Bystroff and Shao [22]. I-sites is a hidden Markov model (HMM) method designed to detect strong relationships between sequence and structure as defined by a library of local structure–sequence relationships. One potential advantage of this method is that the I-sites method is not constrained to fragments of a fixed length (Rosetta is constrained to three- and nine-length fragments) [23]. Thus larger patterns of local structure bias are expected to be detected better by this method. Karplus and coworkers also use a similar approach to detecting fragments of local structure (a two-stage HMM) as part of their *de novo* method [53]. These methods have the primary advantage of better performance when local sequence–structure bias is high (e.g. when local structure is strongly and/or uniquely determined by sequence).

2.5 Clustering as a Heuristic Approach to Approximating Entropic Determinants of Protein Folding

Several protein structure prediction methods are effectively two-step procedures involving the generation of large ensembles of conformations (each being the result of a minimization or simulation) followed by the clustering of the generated ensemble to produce one or more cluster centers that are taken to be the predicted models. Regardless of how one justifies the use of clustering as a means of selecting small numbers of predictions or models from ensembles of decoys conformations, the justification is indirectly supported by the efficacy of the procedure and the resultant observation that clustering has become a central, seemingly required, feature of successful *de novo* prediction methods. Starting with CASP3 the field has witnessed a proliferation of clustering methods as post-simulation processing steps in protein structure prediction methods [14, 51, 96, 108].

Prediction of protein structure *de novo* using Rosetta relies heavily on a final clustering stage. In the first step a large ensemble of potential protein structures is generated, each conformation being the result of an extensive Monte Carlo search designed to minimize the Rosetta scoring function (see Figure 1). We then apply clustering to find the centers of the largest clusters. These cluster centers are ranked by the size of their originating cluster in the ensemble. The tightness of clustering in the ensemble is also used as a measure of method success (larger tighter clusters indicate a higher probability that the method produced correct fold predictions for a given protein). Each Rosetta simulation/Monte Carlo run can be thought of as a fast quench starting from a random point on the conformational landscape (defined by the local structure estimation/fragments). Many of these fast quenches (individual simulations) results in incorrect conformations that score nearly as well as any correct conformations generated in the full ensemble of decoy conformations, as judged by the Rosetta score (a number of other potentials tested also lack discriminative power at this stage). This lack of discrimination by *de novo* scoring functions is partially the result of inaccuracies in the scoring function, limitations in our ability to search the landscape and the fact that entropic terms are a major contributor to the free energy of folding. In any case, this lack of discrimination is mitigated by a final clustering step and it has been shown that the centers of the largest clusters in a clustered Rosetta decoy ensemble are in most cases the conformations closest to native. The ubiquitous use of clustering can be justified in several ways: clustering can be thought of as (i) a heuristic way to approximate the entropy of a given conformation given the full ensemble of decoy conformations generated for a given protein, (ii) a signal averaging procedure, averaging out errors in the low-resolution scoring function, or (ii) taking advantage of foldable-protein specific energy landscape features such as broad energy wells that are the result of proteins evolving to be robust to sequence and conformational changes from the native sequence or structure (a mix of sequence and configurational entropy) [95].

An interesting alternative to the strategy of clustering ensembles of results from independent minimizations is the use of replica exchange methods. Replica exchange methods employ large numbers of simulations spanning a range of temperatures (defined physically if one uses a physical potential or simply as a constant in the exponent of the Boltzmann equation (see Chapter 11) for probabilistic scoring functions). These independent simulations are carried out in parallel and are allowed to exchange temperatures throughout the run. This simulation strategy ideally allows for a random walk in energy space (and thus better sampling) and can be used to calculate entropic term *post facto*. Replica exchange Monte Carlo has been used successfully in the simulation and prediction of protein structure, and is interesting due to its explicit connection to a physical description of the system and its ability to search low energy states without getting trapped [81].

2.6 Balancing Resolution with Sampling, Prospects for Improved Accuracy and Atomic Detail

Every *de novo* structure prediction procedure must strike a delicate balance between the computational efficiency of the procedure and the level of physical detail used to model protein structure within the procedure. Low-resolution models can be used to predict protein topology/folds and sometimes suggest function [15]. Low-resolution models have also been remarkably successful at predicting features of the folding process such as folding rates and phi values [1, 2]. It is clear, however, that modeling proteins (and possibly bound water and other cofactors) at atomic detail and scoring these higher resolution models with physically derived, detailed potentials is a needed development if higher-resolution structure prediction is to be achieved.

Early progress has focused on the use of low-resolution approaches for initially searching the conformational landscape followed by a refinement step where atomic detail and physical scoring functions are used to select and/or generate higher-resolution structures. For example, several studies have illustrated the usefulness of using *de novo* structure prediction methods as part of a two-stage process in which low-resolution methods are used for fragment assembly and the resulting models are refined using a more physical potential and atomic detail (e.g. rotamers) [31] to represent side-chains [18, 71, 102]. In the first step, Rosetta is used to search the space of possible backbone conformations with all side-chains represented as centroids. This process is well described, and has well-characterized error rates and behavior. High-confidence or low-scoring models are then refined using potentials that account for atomic detail such as hydrogen bonding, van der Waals forces and electrostatics.

One major challenge that faces methods attempting to refine *de novo* methods is that the addition of side-chain degrees of freedom combined with the reduced length scale (reduced radius of convergence; one must get much closer to the correct answer before the scoring function recognizes the conformation as correct) of the potentials employed require the sampling of a much larger space of possible conformations. Thus, one has to correctly determine roughly twice the number of bond angles to a higher tolerance if one hopes to succeed. An illustrative example of the difference in length scale (radius of convergence) between low-resolution methods and high-resolution methods is the scoring of hydrogen bonds. In the low-resolution Rosetta procedure backbone hydrogen bonding is scored indirectly by a term designed to pack strands into sheets under the assumption that correct alignment of strands satisfies hydrogen bonds between backbone atoms along the strand and that intra-helix backbone hydrogen bonds are already well accounted for by the local structure fragments. This low-resolution method first reduces strands to vectors, and then scores strand arrangement (and the correct hydrogen bonding implicit in the relative positions/arrangement of all strand vector pairs) via functions dependent on the angular and distance relationships between the two vectors. Thus, the scoring function is robust to a rather large amount of error in the coordinates of individual electron donors and acceptors participating in backbone hydrogen bonds (as large numbers of residues are reduced to the angle and distance between the two vectors representing a given pair of strands). In the high-resolution, refinement mode of Rosetta an empirical hydrogen bond terms with angle and distance dependence between individual electron donors and acceptors is used [88]. This more-detailed hydrogen bond term has a higher fidelity and a more straightforward connection to the calculation of physically realistic energies (meaningful units), but requires more sampling, as smaller changes in the orientation of the backbone can cause large fluctuations in computed energy.

Another major challenge with high-resolution methods is the difficulty of computing accurate potentials for atomic-detail protein modeling in solvent; with electrostatic and solvation terms being among the most difficult terms to accurately model. Full treatment of the free energy of a protein conformation (with correct treatment of dielectric screening) is complicated by the fact that some waters are detectably bound to the surface of proteins and mediate interactions between residues [34]. Another challenge is the computational cost of full treatment of electrostatic free energy by solving the Poisson–Boltzmann or linearized Poisson–Boltzmann equations for large numbers of conformations. In spite of these difficulties several studies have shown that refinement of *de novo* structures with atomic-detail potentials can increase our ability to select and or generate near native structures [78]. These methods can correctly select near native conformations from these ensembles and improve near native structures, but still rely heavily on the initial low-resolution search to produce an ensemble containing good starting structures [63,71,102]. Some recent examples of high-resolution predictions are quite encouraging and an emerging consensus in the field is that higher resolution *de novo* structure prediction (structure predictions with atomic-detail representations of side-chains) will begin to work if sampling is dramatically increased.

Progress in high-resolution structure prediction will invariably be carried out in parallel with methods including, but not limited to, predicting protein–protein interactions, designing proteins and distilling structures from partially assigned experimental data sets. Indeed, many of the scoring and search strategies that high-resolution *de novo* structure refinement methods employ were initially developed in the context of homology modeling and protein design [61,90].

3 Applying Structure Prediction: *De Novo* Structure Prediction in a Systems Biology Context

Sequence databases are growing rapidly, with new genomes being deposited at a phenomenal pace. A large portion of each of these newly sequenced genomes can be expected to contain proteins that have no detectable homologs or only homologs of unknown function. It can be expected that even with the continued progress of large experimental structural biology efforts there will remain a large number of proteins for which *de novo* structure prediction and distant fold recognition methods are the only options.

3.1 Structure Prediction as a Road to Function

The relationship between protein structure and protein function is discussed in detail in Chapter 33, but will be reviewed briefly here in the context of *de novo* structure prediction. One paradigm for predicting the function of proteins of unknown function in the absence of homologs, sometimes referred to as the "sequence-to-structure-to-structure-to-function" paradigm, is based on the assumption that 3-D structure patterns are conserved across a much greater evolutionary distance than recognizable primary sequence patterns [33]. This assumption is based on the results of several structure–function surveys which show that structure similarities (fold matches between different proteins in the PDB) in the absence of sequence similarities imply some shared function in the majority of cases [48,67,70,84,101]. One protocol for predicting protein function based on this observation is to predict the structure of a query sequence of interest and then use the predicted structure to search for fold or structural similarities between the predicted protein structure and experimentally determined protein structures in the PDB or a nonredundant subset of the PDB [49,76,83,85]. There are several problems associated with deriving functional annotation from fold similarity, e.g. old similarities can occur through convergent evolution and thus have no functional implications. Also, aspects of function can change throughout evolution leaving only general function intact across a given fold superfamily [43, 56, 91]. Fold matches between the predicted structures and the PDB are thus treated as sources of putative general functional information, and are functionally interpreted primarily in combination with other methods such as global expression analysis and the predicted protein association network. To circumvent these ambiguities one can (i) use *de novo* structure prediction and/or fold recognition to generate a confidence ranked list of possible structures for proteins or protein domains of unknown function, (ii) search each of the ranked structure predictions against the PDB for fold similarities and possible 3-D motifs, (iii) calculate confidences for the fold predictions and 3-D motif matches, and, finally, (iv) evaluate possible functional roles in the context of the other systems biology data, such as expression analysis, protein interactions, metabolic networks and comparative genomics.

3.2 Initial Application of *De Novo* Structure Prediction

To date there have been few studies using *de novo* structure prediction as a method for genome annotation, due primarily to the computational expense of the calculations and the relative novelty of the methods. These studies have been carried out in combination with a variety of fold recognition and sequence-based methods for gene annotation, and have provided preliminary results that highlight several successes. It is based on these studies that we argue that *de novo* structure prediction is a viable option for exploring genes of unknown function.

The first emergence of *de novo* structure prediction methods for large-scale structure prediction was heavily limited by available computer resources. These studies were essentially pilot studies to evaluate the potential worth of genome-wide *de novo* structure predictions. In one early study workers were limited to generating predictions for 85 proteins in *Mycoplasma genitalium*, producing around 24 correct fold predictions [54]. Another study approached the computational limitation by folding representatives of Pfam protein families of unknown structure and function [14]. Using this method we were able to generate high confidence fold/structure predictions for around 60% of the 510 protein families for which Rosetta predictions were attempted, covering an additional roughly 12% of the sequences available at that time. Subsequent experimental determination for several of these protein families has shown our computed confidence values to be good estimates of our predictive performance, with success rates (rates of correct fold identification) on internal benchmarks and success rates from blind tests (CASP results and recently solved structures) nearly indistinguishable. Alas, the results of this study were not widely used by biologists due partially to the fact that at the time methods for integrating the resultant low-resolution structure predictions with other data types were not in place. The results of these early studies suggested, however, that whole-genome application of *de novo* structure prediction would result in usable annotations if presented to biologists properly, i.e. integrated with other available data types.

3.3 Application on Genome-wide Scale and Examples of Data Integration

Genome-wide measurements of mRNA transcripts, protein concentrations, protein–protein interactions and protein–DNA interactions generate rich sources of data on proteins, both those with known and those with unknown functions [5, 7]. These systems-level measurements seldom suggest a unique function for a given protein of interest, but often suggest their association with or perhaps their direct participation in a previously known cellular process. Investigators using genome-wide experimental techniques are thus routinely generating data for proteins of hitherto unknown function that appear to play pivotal roles in their studies.

The first full-genome application of *de novo* structure prediction was to the genome of *Halobacterium* NRC-1 [12]. This archaeon is an extreme halophile that thrives in saturated brine environments such as the Dead Sea and solar salterns. It offers a versatile and easily assayed system with several well-coordinated physiologies that are necessary for survival in its harsh environment. The completely sequenced genome of *Halobacterium* NRC-1 (containing around 2600 genes) has provided insights into many of its physiological capabilities; however, nearly half of all genes encoded in the halobacterial genome had no known function prior to our re-annotation [29, 30, 79, 93]. A multi-institutional effort is currently underway to study the genome-wide response of *Halobacterium* NRC-1 to its environment, elevating the need for applying improved methods for annotating proteins of unknown function found in the *Halobacterium* NRC-1 genome. Rosetta *de novo* structure prediction was used to predict 3-D structures for 1185 proteins and protein domains (less than 150 residues in length) found in *Halobacterium* NRC-1. Predicted structures were searched against the PDB to identify fold matches [85] and were analyzed in the context of a predicted association network composed of several sources of functional associations, such as predicted protein interactions, predicted operons, phylogenetic profile similarity and domain fusion. This annotation pipeline was also applied to the recently sequenced genome of *Haloarcula marismortui* with similar rates of correct fold identification.

An application of *de novo* structure prediction to yeast has also been described. This study focused on the application and integration of several methods (ranging from experimental methods to *de novo* structure prediction) to 100 essential open reading frames (ORFs) in yeast [47]. For these 100 proteins the group applied affinity purification followed by mass spectrometry (to detect protein binding partners), two-hybrid analysis, florescence microscopy (to localize proteins) and *de novo* structure prediction (Ginzu to separate domains [26, 55] and Rosetta to build structures for domains of unknown function). Due to the cost of experiments and the computational cost of Rosetta *de novo* structure prediction, the group was initially able to prototype the method on just these 100 proteins. Function was assigned to 48 of the proteins (as defined by assignment to Gene Ontology categories). In total, 77 of the 100 proteins were annotated (had confident hits) by on of the methods employed. Given that the starting set represented a difficult set of ORFs of no known function this represents a significant milestone. Scaling this sort of approach up to whole genomes (including large eukaryotic genomes) is still a significant challenge. A grid computing solution (Section 3.4) is currently being employed to complete this study (fold the remaining ORFs in the genome) and, due to the wide use of yeast as a model organism, we can expect this complete resource to be a major step in crossing the social and technical barrier that has so far prevented the wide application of *de novo* structure prediction to biology. A similar approach has also been applied to the Y chromosome of *Homo sapiens* [40]. By integrating fold recognition with *de novo* structure prediction folds were assigned to around 42 of the 60 recognized domains examined (these 60 domains originated from the 27 proteins thought to be encoded on this chromosome at the time of the study). In both of these application, yeast and human, careful thought was put into reducing the set of proteins examined and scaling-up *de novo* structure prediction remains a critical bottleneck (the introduction of all-atom or high-resolution refinement of these predicted structures will only exacerbate this critical need for computing).

3.4 Scaling-up *De Novo* Structure Prediction: Rosetta on the World Community Grid

There are several strategies one can use to limit the number of protein domains for which computationally expensive *de novo* structure prediction needs to be carried out, allowing for the calculation of useful *de novo* structure predictions for only the most relevant subsets of larger genomes, as discussed above. In spite of these strategies, finding the required compute resources has been a constant challenge for the application of *de novo* structure prediction to functional annotation and has limited the application of the method. To circumvent this problem we are currently applying a grid, distributed computing, solution to folding over 100 000 domains with the full Rosetta *de novo* structure prediction protocol (www.worldcommunitygrid.org). These domains were chosen by applying Ginzu [26,55] to over 60 complete genomes as well as several other appropriate sequences in public sequence databases. The results will be integrated with data types that are appropriate/available for a given organism in collaboration with several other groups [12, 47]. This work is ongoing in collaboration with David Baker, Lars Malmstroem (University of Washington) Rick Alther, Bill Boverman and Viktors Berstis (IBM), and United Devices (Austin, TX). Currently (11:10 AM, pacific coast time, 14 April 2005), there are over 1 million volunteers (people who have downloaded the client to run grid-Rosetta) comprising a virtual grid of over 3 million devices. Interested parties wishing to participate (donate idle CPU time on your desktop computer to this project) can download the grid-enabled Rosetta client at www.worldcommunitygrid.org. This amount of computational power will enable us to remove the barrier represented by the computational cost of *de novo* methods.

4 Future Directions

4.1 Structure Prediction and Systems Biology: Data Integration

Even with dramatically improved accuracy we still face challenges due to the ambiguities of the relationship between fold and function seen for many fold families (indeed, even close sequence homology is not always trivial to interpret as functional similarity, see also Chapter 30). Thus, the full potential of *de novo* structure prediction in a systems biology context can only be realized if structure predictions are integrated into larger analysis, and subsequently made accessible to biologists through better data integration, analysis and visualization tools. One clear example of this is provided by the bacterial transcription factors, for which even strong sequence similarity can imply several possible functions and system-wide information is required to determine a meaningful function (the target of a given transcription factor).

4.2 Need for Improved Accuracy and Extending the Reach of *De Novo* Methods

Although I have argued that data integration is as critical a bottleneck as any other and that there are current applications of *de novo* structure prediction, it is clear that improved accuracy is also essential for progress in the field and for the acceptance of *de novo* structure methods by the end users of whole-genome annotations. There is still a significant amount of error in predictions generated using current structure prediction and domain parsing methods. Extending the size limit of protein folding methods is a promising area of active research as is the development of higher-resolution refinement methods. *De novo* structure prediction requires large amounts of CPU time compared to sequence-based and fold recognition methods (although the use of distributed computing and Moore's law continue to make this less of a bottleneck). Integrating *de novo* predictions with orthogonal sources of general and putative functional information, both experimental and computational, will likely facilitate the annotation of significant portions of the protein sequences resulting from ongoing sequencing efforts, as well as proteins in currently sequenced genomes.

Acknowledgments

I would like to thank Viktors Berstis, Rick Alther and Bill Boverman of IBM for their critical work on the World Community Grid, and Phil Bradley, Lars Malmstroem, David Kim, Dylan Chivian and David Baker for myriad comments, opinions, structure prediction, preprints etc.

References

1 Alm, E. and D. Baker. 1999. Matching theory and experiment in protein folding. Curr. Opin. Struct. Biol. **9**: 189–96.

2 Alm, E. and D. Baker. 1999. Prediction of protein-folding mechanisms from free-energy landscapes derived from native structures. Proc. Natl Acad. Sci. USA **96**: 11305–10.

3 Altschul, S. F., T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller and D. J. Lipman. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**: 3389–402.

4 Baker, D. and A. Sali. 2001. Protein structure prediction and structural genomics. Science **294**: 93–6.

5 Baliga N. S., S. J. Bjork, R. Bonneau, M. Pan, C. Iloanusi , M. C. H. Kottemann, L. Hood and J. DiRuggiero. 2004. Systems level insights into the stress response to UV radiation in the halophilic archaeon *Halobacterium* NRC-1. Genome Res. **14**: 1025–35.

6 Baliga, N. S., R. Bonneau, M. T. Facciotti, et al. 2004. Genome sequence of *Haloarcula marismortui*: a halophilic archaeon from the Dead Sea. Genome Res. **14**: 2221–34.

7 Baliga, N. S., M. Pan, Y. A. Goo, et al. 2002. Coordinate regulation of energy transduction modules in *Halobacterium* sp. analyzed by a global systems approach. Proc. Natl Acad. Sci. USA **99**: 14913–8.

8 Bateman, A., E. Birney, R. Durbin, S. R. Eddy, K. L. Howe and E. L. Sonnhammer. 2000. The Pfam protein families database. Nucleic Acids Res. **28**: 263–6.

9 Bateman, A., L. Coin, R. Durbin, et al. 2004. The Pfam protein families database. Nucleic Acids Res. **32**: D138–41.

10 Blanco, F. J., G. Rivas and L. Serrano. 1994. A short linear peptide that folds into a native stable beta-hairpin in aqueous solution. Nat. Struct. Biol. **1**: 584–90.

11 Bonneau, R. and D. Baker. 2001. *Ab initio* protein structure prediction: progress and prospects. Annu. Rev. Biophys. Biomol. Struct. **30**: 173–89.

12 Bonneau, R., N. S. Baliga, E. W. Deutsch, P. Shannon and L. Hood. 2004. Comprehensive *de novo* structure prediction in a systems-biology context for the archaea *Halobacterium* sp. NRC-1. Genome Biol. **5**: R52.

13 Bonneau, R., C. E. Strauss and D. Baker. 2001. Improving the performance of Rosetta using multiple sequence alignment information and global measures of hydrophobic core formation. Proteins **43**: 1–11.

14 Bonneau, R., C. E. Strauss, C. A. Rohl, D. Chivian, P. Bradley, L. Malmstrom, T. Robertson and D. Baker. 2002. *De novo* prediction of three-dimensional structures for major protein families. J. Mol. Biol. **322**: 65–78.

15 Bonneau, R., J. Tsai, I. Ruczinski and D. Baker. 2001. Functional inferences from blind *ab initio* protein structure predictions. J. Struct. Biol. **134**: 186–90.

16 Bonneau, R., J. Tsai, I. Ruczinski, D. Chivian, C. Rohl, C. E. Strauss and D. Baker. 2001. Rosetta in CASP4: progress in *ab initio* protein structure prediction. Proteins **Suppl. 5**: 119–26.

16a Bowers, P. M., C. E. Strauss, and D. Baker. 2000. *De novo* protein
using sparse
NMR data. J. Biomol. NMR **18 (4)**: 311–8.

17 Bradley, P., K. M. S. Misura and D. Baker. 2006. Toward high-resolution *de novo* structure prediction for small proteins. Science **309**: 1868–71.

18 Bradley, P., D. Chivian, J. Meiler, et al. 2003. Rosetta predictions in CASP5: successes, failures, and prospects for complete automation. Proteins **53** (**Suppl. 6**): 457–68.

19 Brenner, S. E., C. Chothia and T. J. Hubbard. 1998. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. Proc. Natl Acad. Sci. USA **95**: 6073–8.

20 Bujnicki, J. M., A. Elofsson, D. Fischer and L. Rychlewski. 2001. LiveBench-1: continuous benchmarking of protein structure prediction servers. Protein Sci. **10**: 352–61.

21 Bystroff, C. and D. Baker. 1998. Prediction of local structure in proteins using a library of sequence–structure motifs. J. Mol. Biol. **281**: 565–77.

22 Bystroff, C. and Y. Shao. 2002. Fully automated *ab initio* protein structure prediction using I-SITES, HMMSTR and ROSETTA. Bioinformatics **18** (**Suppl. 1**): S54–61.

23 Bystroff, C., V. Thorsson and D. Baker. 2000. HMMSTR: a hidden Markov model for local sequence–structure correlations in proteins. J. Mol. Biol. **301**: 173–90.

24 Callihan, D. E. and T. M. Logan. 1999. Conformations of peptide fragments from the FK506 binding protein: comparison with the native and urea-unfolded states. J. Mol. Biol. **285**: 2161–75.

25 Chevalier, B. S., T. Kortemme, M. S. Chadsey, D. Baker, R. J. Monnat and B. L. Stoddard. 2002. Design, activity, and structure of a highly specific artificial endonuclease. Mol. Cells **10**: 895–905.

26 Chivian, D., D. E. Kim, L. Malmstrom, et al. 2003. Automated prediction of CASP-5 structures using the Robetta server. Proteins **53** (**Suppl. 6**): 524–33.

27 Chivian, D., T. Robertson, R. Bonneau and D. Baker. 2003. *Ab initio* methods. Methods Biochem. Anal. **44**: 547–57.

28 Deshpande, N., K. J. Addess, W. F. Bluhm, et al. 2005. The RCSB Protein Data Bank: a redesigned query system and relational database based on the mmCIF schema. Nucleic Acids Res. **33**: D233–7.

29 Devos, D. and A. Valencia. 2001. Intrinsic errors in genome annotation. Trends Genet. **17**: 429–31.

30 Devos, D. and A. Valencia. 2000. Practical limits of function prediction. Proteins **41**: 98–107.

31 Dunbrack, R. L., Jr. and F. E. Cohen. 1997. Bayesian statistical analysis of protein side-chain rotamer preferences. Protein Sci. **6**: 1661–81.

32 Fetrow, J. S., A. Giammona, A. Kolinski and J. Skolnick. 2002. The protein folding problem: a biophysical enigma. Curr. Pharm. Biotechnol. **3**: 329–47.

33 Fetrow, J. S. and J. Skolnick. 1998. Method for prediction of protein function from sequence using the sequence-to-structure-to-function paradigm with application to glutaredoxins/thioredoxins and T1 ribonucleases. J. Mol. Biol. **281**: 949–68.

34 Finney, J. L. 1977. The organization and function of water in protein crystals. Philos. Trans. R. Soc. Lond. B **278**: 3–32.

35 Fischer, D., C. Barret, K. Bryson, et al. 1999. CAFASP-1: critical assessment of fully automated structure prediction methods. Proteins **Suppl. 3**: 209–17.

36 Fischer, D., L. Rychlewski, R. L. Dunbrack, Jr., A. R. Ortiz and A. Elofsson. 2003. CAFASP3: the third critical assessment of fully automated structure prediction methods. Proteins **53** (**Suppl. 6**): 503–16.

37 Galzitskaya, O. V. and B. S. Melnik. 2003. Prediction of protein domain boundaries from sequence alone. Protein Sci. **12**: 696–701.

38 George, R. A. and J. Heringa. 2002. SnapDRAGON: a method to delineate protein structural domains from sequence data. J. Mol. Biol. **316**: 839–51.

39 Ginalski, K., N. V. Grishin, A. Godzik and L. Rychlewski. 2005. Practical lessons from protein structure prediction. Nucleic Acids Res. **33**: 1874–91.

40 Ginalski, K., L. Rychlewski, D. Baker and N. V. Grishin. 2004. Protein structure prediction for the male-specific region of the human Y chromosome. Proc. Natl Acad. Sci. USA **101**: 2305–10.

41 Gray, J. J., S. Moughon, C. Wang, O. Schueler-Furman, B. Kuhlman, C. A. Rohl and D. Baker. 2003. Protein–protein docking with simultaneous optimization of rigid-body displacement and side-chain conformations. J. Mol. Biol. **331**: 281–99.

42 Gray, J. J., S. E. Moughon, T. Kortemme, O. Schueler-Furman, K. M. Misura, A. V. Morozov and D. Baker. 2003. Protein–protein docking predictions for the CAPRI experiment. Proteins **52**: 118–22.

43 Grishin, N. V. 2001. Fold change in evolution of protein structures. J. Struct. Biol. **134**: 167–85.

44 Gunn, J. R. 1997. Sampling protein conformations using segment libraries and a genetic algorithm. J. Chem. Phys. **106**: 4270.

45 Haas, B. J., J. R. Wortman, C. M. Ronning, et al. 2005. Complete reannotation of the *Arabidopsis* genome: methods, tools, protocols and the final release. BMC Biol. **3**: 7.

46 Han, K. F., C. Bystroff and D. Baker. 1997. Three-dimensional structures and contexts associated with recurrent amino acid sequence patterns. Protein Sci. **6**: 1587–90.

47 Hazbun, T. R., L. Malmstrom, S. Anderson, et al. 2003. Assigning function to yeast proteins by integration of technologies. Mol. Cells **12**: 1353–65.

48 Holm, L. and C. Sander. 1997. Dali/FSSP classification of three-dimensional protein folds. Nucleic Acids Res. **25**: 231–4.

49 Holm, L. and C. Sander. 1993. Protein structure comparison by alignment of distance matrices. J. Mol. Biol. **233**: 123–38.

50 Huang, E. S., R. Samudrala and B. H. Park. 2000. Scoring functions for *ab initio* protein structure prediction. Methods Mol. Biol. **143**: 223–45.

51 Hung, L. H. and R. Samudrala. 2003. PROTINFO: secondary and tertiary protein structure prediction. Nucleic Acids Res. **31**: 3296–9.

52 Jones, D. T. 1999. Protein secondary structure prediction based on position-specific scoring matrices. J. Mol. Biol. **292**: 195–202.

53 Karplus, K., R. Karchin, J. Draper, J. Casper, Y. Mandel-Gutfreund, M. Diekhans and R. Hughey. 2003. Combining local-structure, fold-recognition, and new fold methods for protein structure prediction. Proteins **53** (**Suppl. 6**): 491–6.

54 Kihara, D., Y. Zhang, H. Lu, A. Kolinski and J. Skolnick. 2002. *Ab initio* protein structure prediction on a genomic scale: application to the *Mycoplasma genitalium* genome. Proc. Natl Acad. Sci. USA **99**: 5993–8.

55 Kim, D. E., D. Chivian and D. Baker. 2004. Protein structure prediction and analysis using the Robetta server. Nucleic Acids Res. **32**: W526–31.

56 Kinch, L. N. and N. V. Grishin. 2002. Evolution of protein structures and functions. Curr. Opin. Struct. Biol. **12**: 400–8.

57 Kortemme, T., L. A. Joachimiak, A. N. Bullock, A. D. Schuler, B. L. Stoddard and D. Baker. 2004. Computational redesign of protein–protein interaction specificity. Nat. Struct. Mol. Biol. **11**: 371–9.

58 Kortemme, T., A. V. Morozov and D. Baker. 2003. An orientation-dependent hydrogen bonding potential improves prediction of specificity and structure for proteins and protein–protein complexes. J. Mol. Biol. **326**: 1239–59.

59 Kuhlman, B. and D. Baker. 2000. Native protein sequences are close to optimal for their structures. Proc. Natl Acad. Sci. USA **97**: 10383–8.

60 Kuhlman, B., G. Dantas, G. C. Ireton, G. Varani, B. L. Stoddard and D. Baker. 2003. Design of a novel globular protein fold with atomic-level accuracy. Science **302**: 1364–8.

61 Kuhlman, B., J. W. O'Neill, D. E. Kim, K. Y. Zhang and D. Baker. 2002. Accurate computer-based design of a new backbone conformation in the second turn of protein L. J. Mol. Biol. **315**: 471–7.

62 Lazaridis, T. and M. Karplus. 1999. Discrimination of the native from misfolded protein models with an energy function including implicit solvation. J. Mol. Biol. **288**: 477–87.

63 Lee, M. R., J. Tsai, D. Baker and P. A. Kollman. 2001. Molecular dynamics in the endgame of protein structure prediction. J. Mol. Biol. **313**: 417–30.

64 Lesk, A. M. 1997. CASP2: report on *ab initio* predictions. Proteins **Suppl. 1**: 151–66.

65 Lesk, A. M., L. Lo Conte and T. J. Hubbard. 2001. Assessment of novel fold targets in CASP4: predictions of three-dimensional structures, secondary structures, and interresidue contacts. Proteins **Suppl. 5**: 98–118.

66 Liu, J. and B. Rost. 2004. CHOP: parsing proteins into structural domains. Nucleic Acids Res. **32**: W569–71.

67 Lo Conte, L., B. Ailey, T. J. Hubbard, S. E. Brenner, A. G. Murzin and C. Chothia. 2000. SCOP: a structural classification of proteins database. Nucleic Acids Res. **28**: 257–9.

68 Lo Conte, L., S. E. Brenner, T. J. Hubbard, C. Chothia and A. G. Murzin. 2002. SCOP database in 2002: refinements accommodate structural genomics. Nucleic Acids Res. **30**: 264–7.

69 Marqusee, S., V. H. Robbins and R. L. Baldwin. 1989. Unusually stable helix formation in short alanine-based peptides. Proc. Natl Acad. Sci. USA **86**: 5286–90.

70 Martin, A. C., C. A. Orengo, E. G. Hutchinson, et al. 1998. Protein folds and functions. Structure **6**: 875–84.

71 Misura, K. M. and D. Baker. 2005. Progress and challenges in high-resolution refinement of protein structure models. Proteins **59**: 15–29.

72 MOODIE, S. L., J. B. MITCHELL AND J. M. THORNTON. 1996. Protein recognition of adenylate: an example of a fuzzy recognition template. J. Mol. Biol. **263**: 486–500.

73 MOULT, J., T. HUBBARD, K. FIDELIS AND J. T. PEDERSEN. 1999. Critical assessment of methods of protein structure prediction (CASP): round III. Proteins **Suppl. 3**: 2–6.

74 MUNOZ, V. AND L. SERRANO. 1996. Local versus nonlocal interactions in protein folding and stability – an experimentalist's point of view. Fold. Des. **1**: R71–7.

75 MURZIN, A. G. 1999. Structure classification-based assessment of CASP3 predictions for the fold recognition targets. Proteins **Suppl. 3**: 88–103.

76 MURZIN, A. G., S. E. BRENNER, T. HUBBARD AND C. CHOTHIA. 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. **247**: 536–40.

77 NERIA, E., S. FISCHER AND M. KARPLUS. 1996. Simulation of activation free energies in molecular systems. J. Chem. Phys. **105**: 1902.

78 NEVES-PETERSEN, M. T. AND S. B. PETERSEN. 2003. Protein electrostatics: a review of the equations and methods used to model electrostatic equations in biomolecules – applications in biotechnology. Biotechnol. Annu. Rev. **9**: 315–95.

79 NG, W. V., S. P. KENNEDY, G. G. MAHAIRAS, et al. 2000. Genome sequence of *Halobacterium* species NRC-1. Proc. Natl Acad. Sci. USA **97**: 12176–81.

80 NIELSEN, H., S. BRUNAK AND G. VON HEIJNE. 1999. Machine learning approaches for the prediction of signal peptides and other protein sorting signals. Protein Eng. **12**: 3–9.

81 OKAMOTO, Y. 2004. Generalized-ensemble algorithms: enhanced sampling techniques for Monte Carlo and molecular dynamics simulations. J. Mol. Graph. Model. **22**: 425–39.

82 ORENGO, C. A., J. E. BRAY, T. HUBBARD, L. LOCONTE AND I. SILLITOE. 1999. Analysis and assessment of *ab initio* three-dimensional prediction, secondary structure, and contacts prediction. Proteins **Suppl. 3**: 149–70.

83 ORENGO, C. A., F. M. PEARL AND J. M. THORNTON. 2003. The CATH domain structure database. Methods Biochem. Anal. **44**: 249–71.

84 ORENGO, C. A., A. E. TODD AND J. M. THORNTON. 1999. From protein structure to function. Curr. Opin. Struct. Biol. **9**: 374–82.

85 ORTIZ, A. R., C. E. STRAUSS AND O. OLMEA. 2002. MAMMOTH (matching molecular models obtained from theory): an automated method for model comparison. Protein Sci. **11**: 2606–21.

86 OUZOUNIS, C. A. AND P. D. KARP. 2002. The past, present and future of genome-wide re-annotation. Genome Biol. **3**: COMMENT2001.

87 PIEPER, U., N. ESWAR, A. C. STUART, V. A. ILYIN AND A. SALI. 2002. MODBASE, a database of annotated comparative protein structure models. Nucleic Acids Res. **30**: 255–9.

88 ROHL, C. A. 2005. Protein structure estimation from minimal restraints using Rosetta. Methods Enzymol. **394**: 244–60.

89 ROHL, C. A., C. E. STRAUSS, D. CHIVIAN AND D. BAKER. 2004. Modeling structurally variable regions in homologous proteins with Rosetta. Proteins **55**: 656–77.

90 ROHL, C. A., C. E. STRAUSS, K. M. MISURA AND D. BAKER. 2004. Protein structure prediction using Rosetta. Methods Enzymol. **383**: 66–93.

91 ROST, B. 1997. Protein structures sustain evolutionary drift. Fold. Des. **2**: S19–24.

92 ROST, B. AND V. A. EYRICH. 2001. EVA: large-scale analysis of secondary structure prediction. Proteins **Suppl. 5**: 192–9.

93 ROST, B. AND A. VALENCIA. 1996. Pitfalls of protein sequence analysis. Curr. Opin. Biotechnol. **7**: 457–61.

94 RYCHLEWSKI, L. AND D. FISCHER. 2005. LiveBench-8: the large-scale, continuous assessment of automated protein structure prediction. Protein Sci. **14**: 240–5.

95 Shortle, D., K. T. Simons and D. Baker. 1998. Clustering of low-energy conformations near the native structures of small proteins. Proc. Natl Acad. Sci. USA **95**: 11158–62.

96 Simons, K. T., R. Bonneau, I. Ruczinski and D. Baker. 1999. *Ab initio* protein structure prediction of CASP III targets using ROSETTA. Proteins **Suppl. 3**: 171–6.

97 Simons, K. T., C. Kooperberg, E. Huang and D. Baker. 1997. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. J. Mol. Biol. **268**: 209–25.

98 Simons, K. T., I. Ruczinski, C. Kooperberg, B. A. Fox, C. Bystroff and D. Baker. 1999. Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins. Proteins **34**: 82–95.

99 Sonnhammer, E. L., G. von Heijne and A. Krogh. 1998. A hidden Markov model for predicting transmembrane helices in protein sequences. Proc. ISMB **6**: 175–82.

100 Tatusov, R. L., N. D. Fedorova, J. J. Jackson, et al. 2003. The COG database: an updated version includes eukaryotes. BMC Bioinformatics **4**: 41.

101 Todd, A. E., C. A. Orengo and J. M. Thornton. 2001. Evolution of function in protein superfamilies, from a structural perspective. J. Mol. Biol. **307**: 1113–43.

102 Tsai, J., R. Bonneau, A. V. Morozov, B. Kuhlman, C. A. Rohl and D. Baker. 2003. An improved protein decoy set for testing energy functions for protein structure prediction. Proteins **53**: 76–87.

103 Wallace, A. C., R. A. Laskowski and J. M. Thornton. 1996. Derivation of 3D coordinate templates for searching structural databases: application to Ser–His–Asp catalytic triads in the serine proteinases and lipases. Protein Sci. **5**: 1001–13.

104 Ward, J. J., L. J. McGuffin, K. Bryson, B. F. Buxton and D. T. Jones. 2004. The DISOPRED server for the prediction of protein disorder. Bioinformatics **20**: 2138–9.

105 Wedemeyer, W. J. and D. Baker. 2003. Efficient minimization of angle-dependent potentials for polypeptides in internal coordinates. Proteins **53**: 262–72.

106 Yi, Q., C. Bystroff, P. Rajagopal, R. E. Klevit and D. Baker. 1998. Prediction and structural characterization of an independently folding substructure in the src SH3 domain. J. Mol. Biol. **283**: 293–300.

107 Zhang, B., L. Rychlewski, K. Pawlowski, J. S. Fetrow, J. Skolnick and A. Godzik. 1999. From fold predictions to function predictions: automation of functional site conservation analysis for functional genome predictions. Protein Sci. **8**: 1104–15.

108 Zhang, Y. and J. Skolnick. 2004. Tertiary structure predictions on a comprehensive benchmark of medium to large size proteins. Biophys. J. **87**: 2647–55.

13
Structural Genomics

Philip E. Bourne and Adam Godzik

1 Overview

1.1 What is Structural Genomics?

Inspired by the success of the genome-sequencing projects, particularly the Human Genome Project [41], research-funding bodies in the US, Japan and Europe decided to embark on an equally ambitious project of large-scale macromolecular structure determination. Looking at biology in terms of increasing complexity and scale, this made sense – from the sequence of genomes comes structure from which molecular function can be derived. Individual functions define processes that occur in different parts of the cell, different cell types make up an organism and so on. Thus, the next logical step in understanding living systems was large-scale macromolecular structure determination. These efforts, weakly correlated and distributed over several institutions on several continents, became collectively known as structural genomics. The hope was that structural genomics would continue an explosive growth in raw data, knowledge and technology [32,35]. This chapter describes structural genomics on its fifth anniversary. Where representative examples of the work being undertaken are needed, they are taken from the Joint Center for Structural Genomics (JCSG), one of the US structural genomics centers that is close to the authors both in space (it is located in San Diego) and in spirit (one of us, A. G., leads the bioinformatics core at JCSG).

1.2 What are the Motivators?

Whereas the goal of the human genome project was straightforward, i.e. determine the 3 billion nucleotides that comprise the specific genome (human), the goal of structural genomics is less so. The most often stated goal was to provide "structural coverage" of protein space, by solving enough structures that all known proteins could be accurately modeled [7, 42]. Other goals, such as targeting disease-relevant genes, were also listed [8]. Last, but

Bioinformatics – From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

not least, structural genomics aimed to develop new structure determination technologies that would lower the costs and time needed to solve protein structures. Without a formal definition of its goals, structural genomics was adopted as a broad research goal by a loose coalition of researchers from around the world. A list of the current projects is given in Table 1 and up-to-date information is available from the Research Collaboratory for Structural Bioinformatics (RCSB) Protein Data Bank (PDB) [4] which tracks all projects (http://sg.pdb.org/target_centers.html) [10,22].

In the US, structural genomics efforts resulted in the launch of the National Institutes of Health (NIH) Protein Structure Initiative (PSI), which in time developed its own goals and milestones [30] that partly overlapped the original overall goals of structural genomics. The same could be said of developments in other parts of the world. Recently, the PSI initiatives in the US have received their second round of 5-year funding and goals and milestones are being further refined. Table 1 indicates major US centers from round 1 as PSI-1 and the subset of those with major funding in round 2 as PSI-2. Regardless of the stated objectives, structural genomics already accomplished (or perhaps coincided with) a major paradigm shift in structural biology – moving from a strictly functionally driven endeavor to a genomically driven endeavor. As we discuss subsequently, this both requires and is driven by contributions from a variety of communities. From the perspective of bioinformatics research, this includes problems in defining the universe of protein structures, recognition of natural units of protein evolution (domains), understanding the complex relationship between protein function(s) and sequence and structure, development of protein structure prediction in general, and homology modeling in particular, and many others.

1.2.1 Fold Coverage as a Motivator

The often repeated goal of structural genomics is "coverage of protein structural space". However, there are at least two levels on which this goal can be achieved. On one level, solving at least one example of all possible folds would provide some information about how large and complex is protein fold space. On another level, many structures from each fold would have to be solved to provide structural templates for possible comparative modeling for every existing protein. The first level, coarse-grained coverage of fold space, seems to be tractable in terms of number of structures that need to be solved, with a total number of folds estimated to be in the 5000–10 000 range. However, choosing structures that would have to be solved would be a formidable task, as it is difficult to predict from sequence which proteins would have a new fold – once the sequence identity drops below 25% the relationship to an existing structure may not be detected, yet folds are often the same below 10% sequence identity [31] (see Chapters 10 and 12). This

Table 1 Structural genomics projects worldwide

Project	Major objectives (as described on their website)
Protein structure initiative centers US	
Berkeley Structural Genomics Center (BSGC) PSI-1	two minimal genomes, *Mycoplasma genitalium* and *Mycoplasma pneumoniae*
Center for Eukaryotic Structural Genomics (CESG) PSI-2	variety of eukaryotic targets
Joint Center for Structural Genomics (JCSG) PSI-2	targets from all superkingdoms with emphasis on *T. maritima*, mouse and yeast
Midwest Center for Structural Genomics (MCSG) PSI-2	various
NorthEast Structural Genomics Consortium (NESG) PSI-2	coverage of fold space
New York Structural Genomics Research Consortium (NYSGRC) PSI-2	technology development
Southeast Collaboratory for Structural Genomics (SECSG) PSI-1	*Caenorhabditis elegans*, *Homo sapiens* and homologs from *Pyrococcus furiosus*
Structural Genomics of Pathogenic Protozoa Consortium (SGPP) PSI-1	various diseases
Mycobacterium tuberculosis Structural Genomics Consortium (TB) PSI-1	study of tuberculosis
Structural genomics projects in Europe and environs	
Bacterial Targets at IGS-CNRS, France (BIGS)	rickettsia, ORFan targets from *Escherichia. coli*, antibacterial gene targets
deCode – decode Genetics, Iceland	various diseases
Israel Structural Proteomics Center (ISPC)	various
Marseilles Structural Genomics Program, France (MSGP)	unknown
NWSGC – North West Structural Genomics Centre, UK	tuberculosis
Oxford Protein Production Facility, England (OPPF)	technology development
Protein Structure Factory, Germany (PSF)	various
Structural Proteomics in Europe, England (SPINE)	structures related to human health and disease
Mycobacterium Tuberculosis Structural Proteomics Project, Germany (XMTB)	tuberculosis
Paris-Sud Yeast Structural Genomics, France (YSG)	relevant proteins with homologs in *Schizosaccharomyces pombe*
Structural genomics projects in North America	
Montreal-Kingston Bacterial Structural Genomics Initiative, Canada (BSGI)	various
OCSP – Ontario Centre for Structural Proteomics, Canada	various
Project CyberCell, Canada	various

Table 1 continued

Project	Major objectives (as described on their website)
SGC – Structural Genomics Consortium, Canada, UK	various proteins of medical relevance
Structure 2 Function Project, US (S2F)	*Haemophilus influenzae*
Structural genomics projects in Asia	
KSPRO – Korean Structural Proteomics Research Organization, Korea	*Helicobacter pylori* and human cancer genes
RIKEN Structural Genomics Initiative, Japan (RSGI)	various
SGCGES – Structural Genomics Consortium for Research on Gene Expression System, Japan	proteins associated with protein synthesis

difficulty is illustrated in Figure 1, where the growth in the number of new folds is shown not to have increased significantly since the advent of the structural genomics projects in 2000. The picture is complicated by how one defines a new fold. Figure 1 is based upon the Structure Classification of Proteins (SCOP) [1], but other definitions exist as will be described subsequently. Further, proteins not homologous to already crystallized proteins are significantly more difficult to handle and have lower success rates at almost every stage of the structure determination process. On the second level, covering protein space at a fine-grained level requires a number of potential targets counted in hundreds of thousands, thus making the goals of structural genomics essentially unattainable. It is clear that the practical strategy must steer clear from both these extremes and, as we will discuss below, this is indeed what most structural genomics centers have been doing in practice.

Despite improvements in structure determination technology, it is clear that we have not achieved a breakthrough that could be compared to, for example, what shotgun sequencing did for accelerating genome-sequencing projects. Automation and streamlining have been applied at every step, but overall improvements have been incremental rather than dramatic, and as of today even the best centers do not produce more than 10–15 structures a month at a cost of US$50 000–$60 000 per structure. While this is impressive by the standards of structural biology from even a few years ago, at this rate and cost it is clear that fine-grained coverage of protein structural space is impossible to attain. Therefore, the ultimate goal of structural genomics could only be achieved by combining experimental and theoretical approaches, and improvements in comparative modeling are necessary to improve the convergence radius of successful model building.

Another means of defining a goal for structural genomics is to consider that goal from a biological perspective.

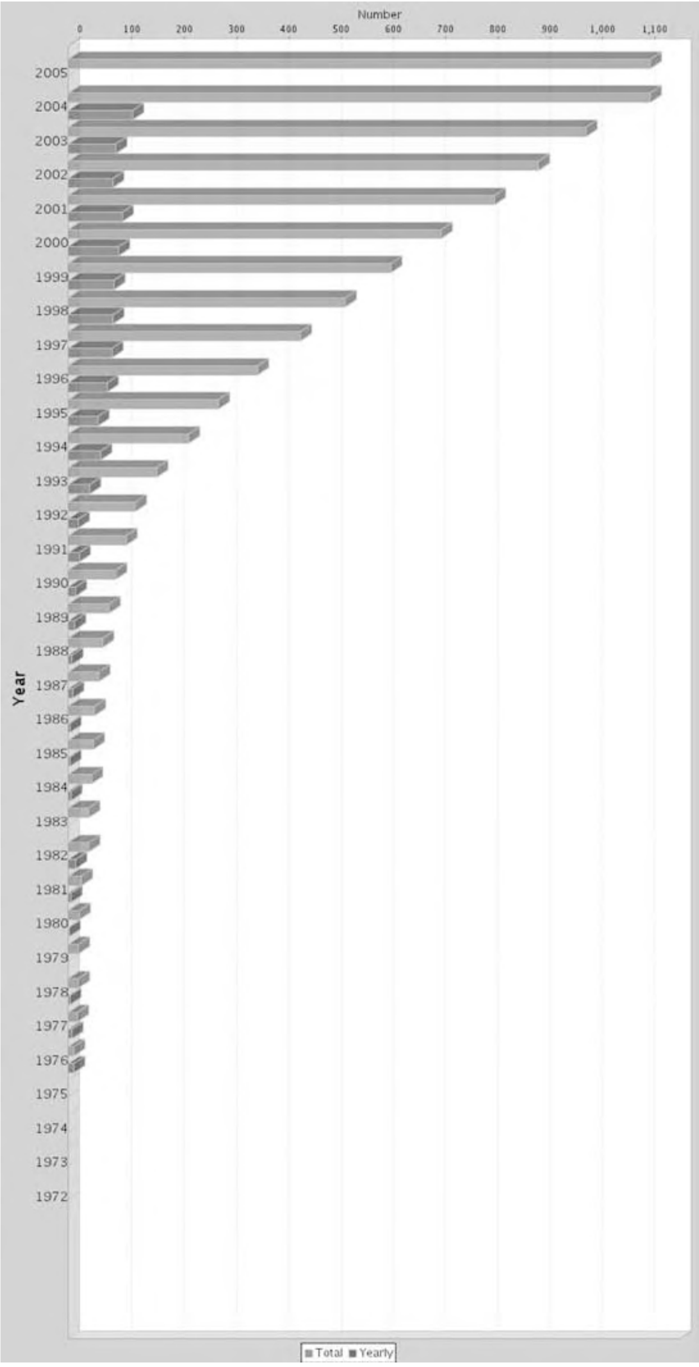


Figure 1 Growth in the number of folds per year as defined by SCOP.

1.2.2 Structural Coverage of an Organism as a Motivator

One stated and ambitious goal of structural genomics is to provide an understanding of an entire organism at the molecular level. Some structural genomics centers focused on single organisms, e.g. the JCSG cloned and attempted the expression of 1777 of the predicted 1877 open reading frames from *Thermotoga maritima* leaving aside some of the putative genes with obvious problems in predicted boundaries, etc. From a predicted 1377 soluble proteins, 705 were expressed and 581 made it to crystallization trials. To date, JCSG have solved 155 *T. maritima* structures, which, when combined with structures of *T. maritima* proteins solved at other centers, gives direct structural coverage of 25% of the expressed soluble proteins and around 12% of this organism's proteome. After taking into account structures that can be modeled through homology and fold recognition, this percentage rises to over 70% (over 90% of predicted crystallizable proteins). Thus, we are only a few dozens structures away from having complete structural coverage of an entire organism. However, as we shall see subsequently, based on a detailed discussion of the coverage of the human genome, what constitutes genome coverage by models is open to interpretation. Beyond that, Brenner and colleagues have pointed out [9] that even the determination of several complete archaeal or bacterial proteomes would still leave many protein families structurally uncovered.

1.2.3 Structure Coverage of Central Metabolism Pathways as a Motivator

Here we consider a specific example. Structures solved at JCSG and other PSI centers allow us to take a structure-based view of the metabolic pathways in *T. maritima*. In collaboration with the SEED project (http://theseed.uchicago.edu/FIG/index.cgi), an integrated *T. maritima* annotation project was initiated combining structural, functional and genomic annotations (Figure 2). Results of this effort, which will soon be available on the *T. maritima* annotation website, will provide a unique genome annotation resource. At this point, all *T. maritima* metabolic pathways can be covered by experimental or modeled structures, providing a first of its kind structural view of an organism's metabolome. Structure determination has resolved many outstanding issues in what seemed to be incomplete or redundant pathways and identified novel aspects of *T. maritima* metabolism. For instance, *T. maritima* is one of only four known organisms that do not depend on biotin decarboxylase for fatty acid metabolism. About 40% of all JCSG-solved *T. maritima* structures had their functional annotations changed or significantly updated after their structures were determined.

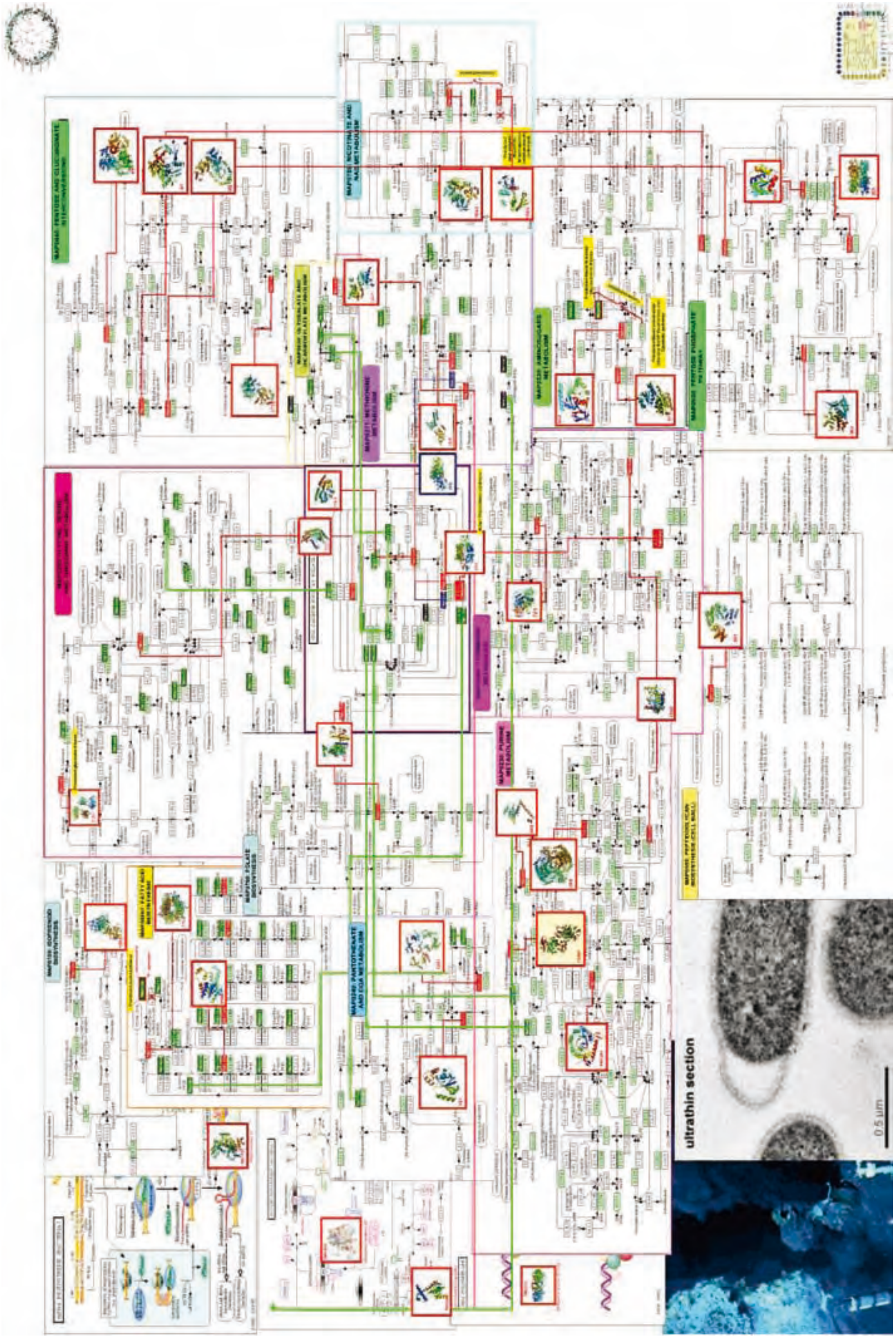
1.2.4 Disease as a Motivator

According to a recent study [45] the PDB currently covers approximately 70% of the human disease categories described by the Online Mendelian Inheritance in Man (OMIM) resource [15], but that coverage is not even. For example, diseases of the central nervous system have a disproportionately large number of solved structures and structural genomics targets relative to the number of proteins associated with this class of disease in the human genome. Blood- and lymph-based diseases have a disproportionately large number of solved structures in the PDB, yet an appropriate underrepresentation of structural genomics targets being attempted. Diseases of the ear, nose and throat, which are currently structurally underrepresented in the PDB, have few targets being attempted and yet there are a significant number of proteins identified as being responsible for this class of disease in the human genome [45]. Some structural genomics projects (Table 1) are targeting specific diseases, e.g. *Mycobacterium tuberculosis*, Chagas' disease and malaria, and a more balanced coverage of proteins associated with human diseases is expected in the next 5 years.

1.3 How Does Structural Genomics Relate to Conventional Structural Biology?

What should be apparent from the above discussion of motivators is that structural genomics changes the conventional paradigm of "I know something about the function of this protein from biochemical evidence, let me determine the single structure of this protein to better elucidate the mechanism" to an almost reverse approach at a different scale of biology, "I see a protein that seems to be important – it is conserved, essential, sits on a virulence island, but its function is unknown, lets solve a structure to start the functional characterization process". One outcome of this paradigm shift is that, for the first time, we are seeing a number of structures which have yet to be functionally characterized offering new challenges for computational biologists to determine function from sequence and structure, not necessarily a trivial undertaking, but success is possible.

Taking a specific example, for 42 of the *T. maritima* structures solved by the JCSG, structural analysis provided strong indications for the possible functions of proteins which were previously listed as "hypothetical proteins". Further, by incorporating structural information into the annotation process, functional annotations of 90 out of 122 structures have been modified, usually by making the function annotation more specific and occasionally by correcting it. Importantly, for some proteins even knowledge of their three-dimensional (3-D) structure did not help to elucidate their function. For instance, TM0875 (a specific JCSG target), with a unique fold and no known homologs, remains uncharacterized a few years after structure determination.


ultrathin section
0.5 µm

Currently, over 900 proteins in PDB are classified as "structural genomics unknown function".

Interestingly, while the value of structural genomics was questioned by some structural biologists at the outset, the consensus opinion now seems to be that advances made in structure determination through structural genomics have fed back to impact conventional structural biology laboratories through, for example, improved software, streamlined procedures at the synchrotron beamlines, and improved techniques for expression, purification and crystallization. We now consider some of these advances.

2 Methodology

Structural genomics employ a variety of methods – X-ray crystallography, nuclear magnetic resonance (NMR), electron microscopy, neutron diffraction, mass spectrometry, etc. It is beyond the scope of this chapter to describe all of these. Rather, we consider X-ray crystallography as the most prevalent of the methods (85% of structures in the PDB to date) to illustrate the impact that structural genomics is having on the methods employed to solve structures by single crystal X-ray diffraction. Figure 3 is a schematic of the basic process. We consider each step and demonstrate what has been achieved by one project, the JCSG, as an example of progress that is being made.

2.1 Target Selection

The motivation for selecting targets was introduced above. The goal of target selection is to ascertain that the sequence of the protein meets the criteria defined for the anticipated structural outcome. That could be biological, i.e. it has a particular function usually determined by identification of homology to another protein known to have this function (see Chapters 30–35), or methodological, i.e. a specific globular domain can be identified which is likely to be amenable to crystallization. Neither recognizing distant homologs nor domains from sequence is a solved problem, although these are active areas of endeavor. See, for example, Ref. [20] for the latest on domain recognition from the Sixth Critical Assessment of Structure Prediction (CASP) [39].

←

Figure 2 Fragment of the metabolic map of *T. maritima* with experimental structures identified by ribbon diagrams and models identified by a green highlight of the enzyme name. Most of the pathways have complete structural coverage and the remaining proteins are being targeted in the fifth year of JCSG.

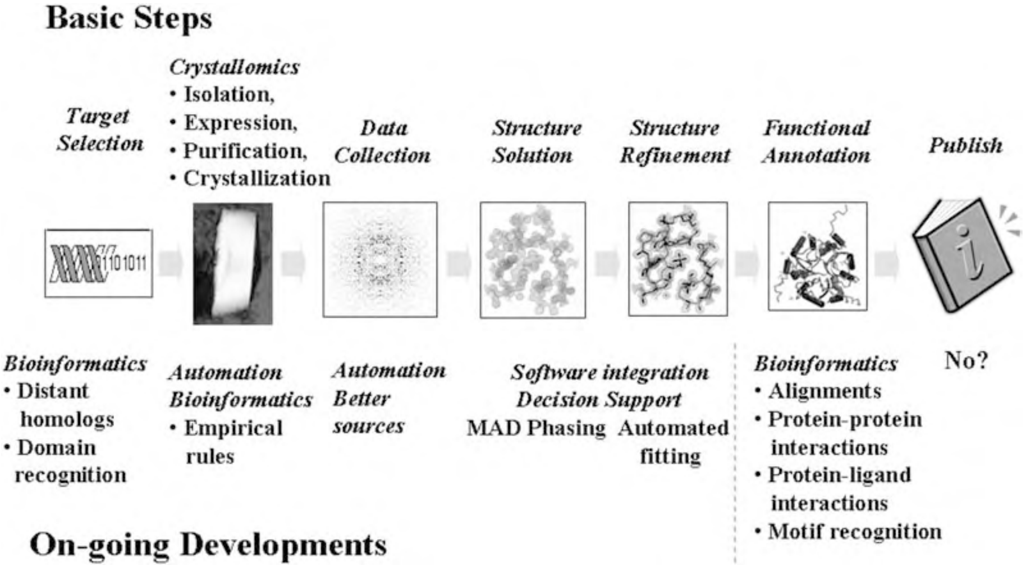


Figure 3 The basic steps in a single-crystal X-ray diffraction experiment (top) and associated on-going developments (bottom) being catalyzed by structural genomics efforts worldwide.

2.2 Crystallomics

Crystallomics is a term to collectively define the steps of protein isolation, expression, purification and crystallization. The initial phase of structural genomics has yielded great progress here. Large-scale, fully automated facilities for protein production have been developed by all structural genomics centers and, increasingly, many commercial solutions are available. Structural genomics centers cloned over 56 000 targets, expressed over 30 000 targets, purified over 10 000 targets and crystallized about 4000 targets in the period since program inception in 2000 to October 2005 (Figure 4). Additional constantly updated statistics are available at http://targetdb.pdb.org/. Consider the approach of JCSG. Many options for creating expression systems were evaluated to maximize flexibility and minimize cost. Ultimately the JCSG team chose to automate a conventional cloning approach. They developed a robotic platform and were able to provide up to 384 validated clones per week. To date, over 2500 clones have been generated and expressed with this system from over 30 000 attempts. The system is efficient, needing only a single person to operate. *Escherichia coli* has been used as the expressions system. To purify the expressed protein, a two-process system has been adopted, both processes being controlled by robotics. Together, they can produce 48–96

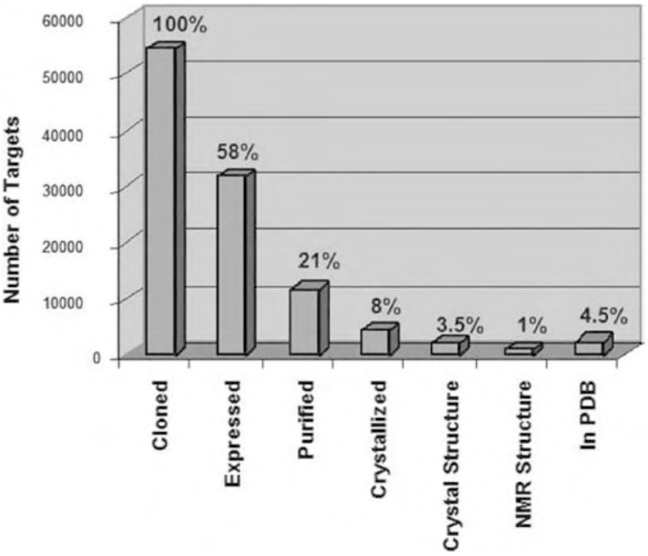


Figure 4 Progress within the structural genomics initiative worldwide as of 17 November 2005.

proteins a week on a 10–50 mg scale. Further details on this process can be found at http://www.jcsg.org/scripts/prod/technologies1.html.

Crystallization strategies vary but are high throughput involving multiwell robotic systems. The JCSG system includes an automated system for shooting digital images of each well and automatic recognition of crystalline samples. To date over 3 million images have been taken and analyzed. Once a sample is identified a prescreening is undertaken to determine if the crystal is of diffraction quality before being sent to the synchrotron for data collection. In excess of 500 crystals are prescreened by JCSG on a monthly basis.

2.3 Data Collection

Crystal samples are transported to the synchrotron in specially designed compact, cylindrical, aluminum crystal cassettes, which holds 96 crystals. Upon reaching the synchrotron samples are automatically mounted, entered into the high-energy X-ray beam and aligned automatically, a process taking approximately 30 s. Using a video system it is intended that data collection will eventually be done remotely without the need for the researcher to travel to the beamline itself. The online report of each crystalline sample is automatically augmented with information from the data collection process.

2.4 Structure Solution

Data collection provides X-ray diffraction patterns from the ordered crystal lattice which appear as a set of diffraction spots. The positions of the spots is defined by the size and type of crystal lattice; however, the intensity of the spots is a function of their amplitude based on the types of atoms (known) and their phases based on the positions of the atoms (unknown at the outset). This is referred to as the phase problem in crystallography. The majority of structures determined by the structure genomics centers worldwide solve the phase problem by establishing a starting set of phases using multi-wavelength anomalous dispersion [17] which requires collecting data at slightly different wavelengths, a process that is possible using synchrotron beamlines which can be tuned for this purpose. The discreet scattering of the X-ray beam by the electron cloud from specific atom types (anomalous scatterers), usually selenium introduced into the structure in the form of selenomethionine, when collected at different wavelengths, provides a starting set of phases, based on the atomic positions of the anomalous scatterers which can now be resolved. A disadvantage of this approach is that data must be collected at each wavelength, lengthening the data collection time. An alternative approach to establishing a starting set of phases is to have a model which approximates the final atomic positions. From this model a starting set of phases can be derived, hence solving the phase problem assuming, first, that the model is accurate enough and, second, that it can be positioned correctly within the crystal lattice. This technique, known as molecular replacement (MR), works well for similar structures, e.g. taking an existing solved native protein structure and using it to phase a mutant structure containing post-translational modifications. An exciting prospect is to routinely use a theoretically derived model structure to determine a staring set of phases. Likelihood for the routine success of this approach relies on having a wide variety of experimentally determined structures from which to model. While this represents somewhat of a chicken-and-egg situation, the number of available structures is increasing rapidly.

In general, MR solutions are seldom attempted (and are even less often successful) against templates with less than 35% sequence identity. Using a fold recognition approach [13] as opposed to an approach based solely on homology, to date, the JCSG MR pipeline was successfully applied to over 20 cases with less than 35% sequence identity, 10 cases with less than 30% and several cases where sequence identity was close to 15%. The analysis shows that fold recognition models, derived from work done in the discipline of protein structure prediction [44], have a significantly higher success rate than homology modeling, especially when the unknown structure and the search model share less than 35% sequence identity. Using the software programs

MOLREP [40] and EPMR [21], three out of 26 MR targets under 35% sequence identity could only be solved with models derived from fold recognition methods, and six showed significantly better statistics and behavior in subsequent refinement [34] than those defined by homology.

2.5 Structure Refinement

Structure refinement implies maximizing the agreement between the intensities observed on the diffraction pattern spots and those calculated from the atomic model. This can be roughly divided into two tasks: (i) getting the main chain and side chains into an optimal position and (ii) performing a final minimization. Ideally both steps must be completely automated to maximize throughput. Improvements to algorithms and usability of software are key factors in this process. A final check must be made to be sure the stereochemical quality of the model is reasonable and this is done primary by checking the relatively low-resolution structure of the complete macromolecule against what is known about macromolecular structure *en masse* (e.g. allowable dihedral angles) and what is known from high-resolution structures of small molecules, such as single amino acids and nucleotides. Despite significant progress in recent years, large parts of the refinement process are still done by hand and high-quality refinement is one of the most time-consuming tasks in high-throughput structure determination. Despite these difficulties, the quality of structures determined by structural genomics centers matches and often exceeds the quality of structures coming from standard structural biology laboratories [38].

2.6 PDB Deposition

Deposition within the PDB (http://www.pdb.org) [4] is a requirement for all structural genomics centers, thereby facilitating the maintenance of a single worldwide archive of macromolecular structures. An ideal goal is to fully automate the deposition process whereby a full structure submission contains not just the final atomic coordinates and experimental data (NMR restraints or X-ray structure factors), but all experimental information including experimental protocols for all the above steps. This would then be validated by the PDB prior to submission. This process is not yet in place, but significant progress has been made. A key element of the process is an ontology which defines in a formal way the items to be collected and their interrelationships. While details of the structure itself have been defined in this way using an ontology referred to as the macromolecular crystallographic information file (mmCIF) [14], additional ontological terms for the various detailed experimental protocols have yet to be fully defined. The progress that has been made thus far is reflected in the Protein Expression Purification and Crystallization Database (PEPCdb; http://pepcdb.pdb.org) which collects experimental protocols according to the beginnings of a full ontological description.

2.7 Functional Annotation

One of the underling goals of structural genomics is to study the relationship between gene sequence, protein structure and protein function, thereby expanding the knowledge of the underlying biology. However, at present most structural genomics centers, by design, stop after structure determination. As a result, a large number of proteins solved by structural genomics groups are listed in the PDB as "hypothetical proteins". This growing list provides raw data from which bioinformatics groups worldwide can apply a variety of methods to assigning putative function(s) to these uncharacterized structures (see Chapters 30–35, especially Chapter 33) [46]. Here, we outline a few of the approaches that have been adopted. In the results section some success stories are introduced. Popular approaches are as follows.

2.7.1 Biological Multimeric State

The structure solved in many cases does not comprise the biologically active molecule. Rather, it represents a unique component. That component may be one domain in a multi-domain protein, a situation found in the PDB in general. For example, multiple SH2 and SH3 domains have been solved and are known to be part of a larger macromolecular complex. Alternatively, the application of crystallographic symmetry can be used to construct the biologically active molecule. Identifying what components comprise the biologically active molecule often requires expert input, although efforts have been made to automate this determination. The Protein Quaternary Server (PQS) uses the notion of proximity of components to define the multimeric state [18].

2.7.2 Active-site Determination

This is an active area of research (see [37] for a review and Chapter 33). Active sites include a small number of residues involved in catalysis, substrate and cofactor binding sites, sites of protein–protein interaction, phosphorylation sites, glycosylation sites, fatty acylation sites, prosthetic group binding sites, hinge regions, domain–domain contacts, sites of membrane association and more. The complexity and importance of the problem is well illustrated by subtilisin and chrymotrypsin. Both are endopeptidases, yet share no sequence identity and their folds are unrelated. However, they share an identical 3-D motif comprising a Ser–His–Asp catalytic triad. The challenge becomes one of identifying an identical motif in two different structures undoubtedly resulting from convergent evolution. Methods are varied, but all comprise basic steps of protein structure representation, application of a search algorithm and assessment of the reliability of the result. Early work used a graph theoretic approach [2], progressing to the use of fuzzy functional forms [11], template modeling [43] and, most recently, an elegantly simple approach based on the proximity of the active site of an enzyme to the centroid of the molecule [3] has emerged. Each method builds upon empirical observations made as more structures are determined and functions classified by biochemical analysis. Thus, we have a rich repertoire from which functional prediction can only improve.

The need for improved methods for function prediction from structure leads to increased research into automated function prediction. The first two meetings of the Automated Functional Prediction (AFP) special interest group were held in 2005 and 2006 (http://biofunctionprediction.org) and the proceedings will be published in 2006. The first metaserver, collecting and analyzing prediction for several servers is now in beta testing (http://jafa.burnham.org).

2.8 Publishing

The original macromolecular structures represented a scientist's life's work. We are now faced with a situation where the rate-determining steps may well be writing the publication. Hence, a number of structures, while deposited in the PDB, remain unpublished. At the time of writing only a small percentage of structures determined by structural genomics are described by peer-reviewed publications, placing additional emphasis on the individual centers websites and the PDB to disseminate as much information about these structures as possible.

JCSG structures are shared with the scientific community not only through deposition in the PDB, but also through publication of a "structure note". Structure notes are short papers describing the annotation, biology, structure and functional implications of each protein. The process of collecting all relevant data from all stages of the JCSG pipeline has been streamlined through the central JCSG database, which includes information on the sequence, annotation, cloning, purification, crystallization, data collection, structure solution, tracing, refinement and structural evaluation. The structure note automatically captures any functional information in the JCSG annotation system (functional annotation is described above). The paper introduction, for example, includes annotation information, with a brief biological background taken and curated from the Pfam [12], Interpro [28], SwissProt [6], BRENDA [33] and SEED databases (http://theseed.uchicago.edu/FIG/index.cgi). Methodological and experimental data, as well as all crystallographic statistics, are automatically harvested from the JCSG database, and assembled into purification, crystallization, structure solution and refinement paragraphs. The structure description and the preparation of figures are done manually using PyMOL (http://pymol.sourceforge.net/). Structures are analyzed, compared and evaluated for biological significance using a plethora of structure analysis tools including structural homology searches (DALI [19], CE [36], FATCAT [25]) and extensive literature searches. In this way the preparation of a structure note is a semiautomated process.

3 Results – Number and Characteristics of Structures Determined

As of 17 November 2005 there were 90 613 targets under consideration by structural genomics projects. Of these 57 019 had been cloned and could be considered under investigation. Figure 4 from http://targetdb.pdb.org/ [10] shows the success rate for the steps described in Section 2. A total of 2540 structures have appeared in the PDB, which is 4.5% of the targets under investigation; 1% has come from NMR structure determination and 3.5% from X-ray crystallography. At that time 7.5% of all structures in the PDB could be considered from structural genomics, with an overall contribution of between 15 and 20% per year. An earlier study from Todd and coworkers [38], when only 316 structures had been deposited, indicated that the quality and size of structures determined by structural genomics versus functionally driven structure determination were comparable. Further, 29% of the domains solved revealed evolutionary relationships not apparent from sequence alone. Similarly, 19 and 11% contributed new superfamilies and folds, respectively. While this number of folds is significantly higher that the contribution from all structures (2% based on the SCOP definition of fold as indicated earlier), it reflects on the difficulty of finding new folds and surely indicates that protein fold space is indeed quite limited.

To get a sense of what structural genomics is contributing, it is first necessary to get some measure of what structure is contributing overall to our understanding of living systems. Clearly, this contribution is somewhat intangible and can be defined in different ways. One recent approach was to review what both structures and targets contribute to the functional and disease coverage of the human genome [45]. In some sense this measure cuts across the various criteria for choosing structural genomics targets that was outlined above. This contribution was measured by looking at the functional coverage of the human genome using either EC numbers (http://www.chem.qmul.ac.uk/iubmb/enzyme/) or Gene Ontology (GO) classifiers [16] and disease via the OMIM [15] classifiers and comparing what the solved structures and targets contributed. Human genome annotation was taken from Ensembl [5] and structure data from the PDB and targets from targetdb [10], the repository maintained by the RCSB PDB of protein sequences from all the structural genomics centers that are being considered for structure determination (http://targetdb.pdb.org). Comparisons were made for both single domains and whole structures. In addition, the ability to homology model was ascertained based on results from SUPERFAMILY [26]. SUPERFAMILY identifies proteins within complete proteomes based on their structural characterization. As such it represents the percentage of a given proteome that can be modeled by existing structures. The results can be summarized as follows:

- Single domains cover 37% of the GO molecular function classes identified in the human genome
- Whole structures cover 25% of the human genome.
- The 37% domain coverage extends to 56% using homology modeling.
- The 25% whole structure coverage extends to 31% using homology models.
- If all current structural genomics targets were solved (~3 times the current PDB):

  37% goes to 69%
  25% goes to 44%

4 Discussion

4.1 Follow-up Studies

One of the ultimate measures of the impact of a new structure is the number of follow-up studies and publications. This impact may not be apparent for several years and is difficult to assess in this comparatively short time frame. However, taking as an example JCSG, their structures have evoked numerous individual collaboration agreements (over 50), associations with larger consortiums detailed in Section 3.7, and numerous requests for clones and proteins for biochemical studies. As an example, TM0449 [23, 24] (PDB code: 1kg4) represents a novel fold which has inspired studies from three different laboratories [27, 29] and has led to the elucidation of a novel biochemical pathway of thymidylate synthesis present, among others, in several important human pathogens. TM0449 and its homologs present an attractive antibacterial drug target, since humans and most eukaryotes depend upon the conventional thymidylate synthase.

4.2 Examples of Functional Discoveries

Again we use JCSG to illustrate the power of long range function and structure projections. In these cases, a bacterial structure from a relatively obscure organism such as *T. maritima* proves to have significant biomedical relevance. For example, the *T. maritima* protein TM1620 provided a template to model a human protein PA26, which belongs to GADD (genes active in DNA defense) family that is highly upregulated in several cancers. It also highlights an interesting conservation of DNA antioxidant defense from bacteria to humans. TM1620 is the second protein solved in this family (the first was AhpD from *Mycobacterium tuberculosis*), which is closely related to 1300 proteins from all kingdoms of life. The mechanism of action of proteins from this family is not clear.

As a second example, TM0813 was shown to be unexpectedly similar to a domain from a known protein involved in antibiotic resistance. Interestingly, a homologous domain is also present in sacsin – a protein whose mutations are responsible for a human neurogenerative disease resulting in autosomal recessive spastic ataxia, often found in Quebec, but also in Turkey and several other areas of the world. In these and other similar cases there is a chance that bacterial proteins, which are easier to characterize and study, would provide hints as to specific mechanism of action of their (very distant) human homologs.

5 The Future

Remaining PSI centers in the US (labeled PSI-2 in Table 1) have just received a second round of 5-year funding and are working together to define the most valuable target list of proteins to be structurally determined. While the objectives of structural genomics remain relatively nebulous relative to the completion of the human genome, solving a significant number of protein structures on that final target list will have a significant outcome. That outcome will be measured in an improved understanding of structure-function relationships, improved coverage of protein fold space and improved technologies for all concerned within the science of structural biology.

References

1 Andreeva, A., D. Howorth, S. E. Brenner, T. J. Hubbard, C. Chothia and A. G. Murzin. 2004. SCOP database in 2004: refinements integrate structure and sequence family data. Nucleic Acids Res. **32**: D226–9.

2 Artymiuk, P. J., A. R. Poirrette, H. M. Grindley, D. W. Rice and P. Willett.

1994. A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structures. J. Mol. Biol. **243**: 327–44.

3 Ben-Shimon, A. and M. Eisenstein. 2005. Looking at enzymes from the inside out: the proximity of catalytic residues to the molecular centroid can be used for detection of active sites and enzyme–ligand interfaces. J. Mol. Biol. **351**: 309–26.

4 Berman, H. M., J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne. 2000. The Protein Data Bank. Nucleic Acids Res. **28**: 235–42.

5 Birney, E., T. D. Andrews, P. Bevan, et al. 2004. An overview of Ensembl. Genome Res. **14**: 925–8.

6 Boeckmann, B., A. Bairoch, R. Apweiler, et al. 2003. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res. **31**: 365–70.

7 Brenner, S. E. 2001. A tour of structural genomics. Nat Rev Genet **2**: 801–9.

8 Burley, J. 1999. The ethics of therapeutic and reproductive human cloning. Semin. Cell Dev. Biol. **10**: 287–94.

9 Chandonia, J. M. and S. E. Brenner. 2005. Implications of structural genomics target selection strategies: Pfam5000, whole genome, and random approaches. Proteins **58**: 166–79.

10 Chen, L., R. Oughtred, H. M. Berman and J. Westbrook. 2004. TargetDB: a target registration database for structural genomics projects. Bioinformatics **20**: 2860–2.

11 Fetrow, J. S. and J. Skolnick. 1998. Method for prediction of protein function from sequence using the sequence-to-structure-to-function paradigm with application to glutaredoxins/thioredoxins and T1 ribonucleases. J. Mol. Biol. **281**: 949–68.

12 Finn, R. D., J. Mistry, B. Schuster-Bocklerv 2006. Pfam: clans, web tools and services. Nucleic Acids Res. **34**: D247–51.

13 Friedberg, I., L. Jaroszewski, Y. Ye and A. Godzik. 2004. The interplay of fold recognition and experimental structure determination in structural genomics. Curr. Opin. Struct. Biol. **14**: 307–12.

14 Greer, D. S., J. D. Westbrook and P. E. Bourne. 2002. An ontology driven architecture for derived representations of macromolecular structure. Bioinformatics **18**: 1280–1.

15 Hamosh, A., A. F. Scott, J. S. Amberger, C. A. Bocchini and V. A. McKusick. 2005. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic Acids Res. **33**: D514–7.

16 Harris, M. A., J. Clark, A. Ireland, et al. 2004. The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res. **32**: D258–61.

17 Hendrickson, W. A. 1991. Determination of macromolecular structures from anomalous diffraction of synchrotron radiation. Science **254**: 51–8.

18 Henrick, K. and J. M. Thornton. 1998. PQS: a protein quaternary structure file server. Trends Biochem. Sci. **23**: 358–61.

19 Holm, L. and C. Sander. 1997. Dali/FSSP classification of three-dimensional protein folds. Nucleic Acids Res. **25**: 231–4.

20 Kim, D. E., D. Chivian, L. Malmstrom and D. Baker. 2005. Automated prediction of domain boundaries in CASP6 targets using Ginzu and RosettaDOM. Proteins **61 (Suppl. 7)**: 193–200.

21 Kissinger, C. R., D. K. Gehlhaar and D. B. Fogel. 1999. Rapid automated molecular replacement by evolutionary search. Acta Crystallogr. D **55**: 484–91.

22 Kouranov, A., L. Xie, J. de la Cruz, L. Chen, J. Westbrook, P. E. Bourne and H. M. Berman. 2006. The RCSB PDB information portal for structural genomics. Nucleic Acids Res. **34**: D302–5.

23 Kuhn, P., S. A. Lesley, I. I. Mathews, et al. 2002. Crystal structure of thy1, a thymidylate synthase complementing protein from *Thermotoga maritima* at 2.25 Å resolution. Proteins **49**: 142–45.

24 LESLEY, S. A., P. KUHN, A. GODZIK, et al. 2002. Structural genomics of the *Thermotoga maritima* proteome implemented in a high-throughput structure determination pipeline. Proc. Natl Acad. Sci. USA **99**: 11664–11669.

25 LI, Z., Y. YE AND A. GODZIK. 2006. Flexible Structural Neighborhood – a database of protein structural similarities and alignments. Nucleic Acids Res. **34**: D277–80.

26 MADERA, M., C. VOGEL, S. K. KUMMERFELD, C. CHOTHIA AND J. GOUGH. 2004. The SUPERFAMILY database in 2004: additions and improvements. Nucleic Acids Res. **32**: D235–9.

27 MATHEWS, I. I., A. M. DEACON, J. M. CANAVES, D. MCMULLAN, S. A. LESLEY, S. AGARWALLA AND P. KUHN. 2003. Functional analysis of substrate and cofactor complex structures of a thymidylate synthase-complementing protein. Structure **11**: 677–90.

28 MULDER, N. J., R. APWEILER, T. K. ATTWOOD, et al. 2005. InterPro, progress and status in 2005. Nucleic Acids Res. **33**: D201–5.

29 MURZIN, A. G. 2002. Biochemistry. DNA building block reinvented. Science **297**: 61–2.

30 NIGMS. 2005. *Protein Structure Initiative Mission Statement*. National Institute of General Medical Sciences, Bethesda, MD.

31 ROST, B. 1999. Twilight zone of protein sequence alignments. Protein Eng. **12**: 85–94.

32 SALI, A. 1998. 100,000 protein structures for the biologist. Nat. Struct. Biol. **5**: 1029–32.

33 SCHOMBURG, I., A. CHANG, C. EBELING, M. GREMSE, C. HELDT, G. HUHN AND D. SCHOMBURG. 2004. BRENDA, the enzyme database: updates and major new developments. Nucleic Acids Res. **32**: D431–3.

34 SCHWARZENBACHER R., A. GODZIK, S. K. GRZECHNIK AND L. JAROSZEWSKI. 2004. The importance of alignment accuracy for molecular replacement. Acta Crystallgr. D **60**: 1229–36.

35 SHAPIRO, L. AND C. D. LIMA. 1998. The Argonne Structural Genomics Workshop: Lamaze class for the birth of a new science. Structure **6**: 265–7.

36 SHINDYALOV, I. N. AND P. E. BOURNE. 1998. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng. **11**: 739–47.

37 THORNTON, J. M., A. E. TODD, D. MILBURN, N. BORKAKOTI AND C. A. ORENGO. 2000. From structure to function: approaches and limitations. Nat. Struct. Biol. **7 (Suppl.)**: 991–4.

38 TODD, A. E., R. L. MARSDEN, J. M. THORNTON AND C. A. ORENGO. 2005. Progress of structural genomics initiatives: an analysis of solved target structures. J. Mol. Biol. **348**: 1235–60.

39 TRESS, M., I. EZKURDIA, O. GRANA, G. LOPEZ AND A. VALENCIA. 2005. Assessment of predictions submitted for the CASP6 comparative modelling category. Proteins **61 (Suppl. 7)**: 27–45.

40 VAGIN, A. AND A. TEPLYAKOV. 1997. MOLREP: an automated program for molecular replacement. J. Appl. Crystallogr. **30**: 1022–1025.

41 VENTER, J. C., M. D. ADAMS, E. W. MYERS, et al. 2001. The sequence of the human genome. Science **291**: 1304–51.

42 VITKUP, D., E. MELAMUD, J. MOULT AND C. SANDER. 2001. Completeness in structural genomics. Nat. Struct. Biol. **8**: 559–66.

43 WALLACE, A. C., N. BORKAKOTI AND J. M. THORNTON. 1997. TESS: a geometric hashing algorithm for deriving 3D coordinate templates for searching structural databases. Application to enzyme active sites. Protein Sci. **6**: 2308–23.

44 WANG, G., Y. JIN AND R. L. DUNBRACK, JR. 2005. Assessment of fold recognition predictions in CASP6. Proteins **61** (**Suppl. 7**): 46–66.

45 XIE, L. AND P. E. BOURNE. 2005. Functional coverage of the human genome by existing structures, structural genomics targets, and homology models. PLoS Comput. Biol. **1**: e31.

46 YAKUNIN, A. F., A. A. YEE, A. SAVCHENKO, A. M. EDWARDS AND C. H. ARROWSMITH. 2004. Structural proteomics: a tool for genome annotation. Curr. Opin. Chem. Biol. **8**: 42–8.

14
RNA Secondary Structures

Ivo L. Hofacker and Peter F. Stadler

1 Secondary Structure Graphs

1.1 Introduction

The tendency of complementary strands of DNA to form double helices is well known since the work of Watson and Crick. Single-stranded nucleic acid sequences will in general contain many complementary regions that have the potential to form double helices when the molecule folds back onto itself. The resulting pattern of double-helical stretches interspersed with loops is what is called the *secondary structure* of an RNA or DNA. Secondary structure elements may in turn be arranged in space to form three-dimensional (3-D) tertiary structure, leading to additional noncovalent interactions (an example is shown in Figure 1). Energetically, however, these tertiary interactions are weaker than secondary structure. As a consequence RNA folding can be regarded as a hierarchical process in which secondary structure forms before tertiary structure [129, 130]. Since formation of tertiary structure usually does not induce changes in secondary structure, the two processes can be described independently. Functional RNA molecules (tRNAs, rRNAs, etc., as opposed to pure coding sequences), usually have characteristic spatial structures – and therefore also characteristic secondary structures – that are prerequisites for their function. As a consequence, secondary structures are highly conserved in evolution for many classes of RNA molecules.

Both the experimental determination of full spatial RNA structures and computational predictions of RNA 3-D structures are very hard tasks – arguably even harder than the corresponding problems for proteins [62, 82]. Computational approaches to RNA tertiary structure thus have been successful only for selected cases (see Chapter 15). RNA secondary structures, on the other hand, not only have a definite physical meaning as folding intermediates and are useful tools in the interpretation RNA molecules, but they give rise to efficient computational techniques. Secondary structure

Bioinformatics – From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

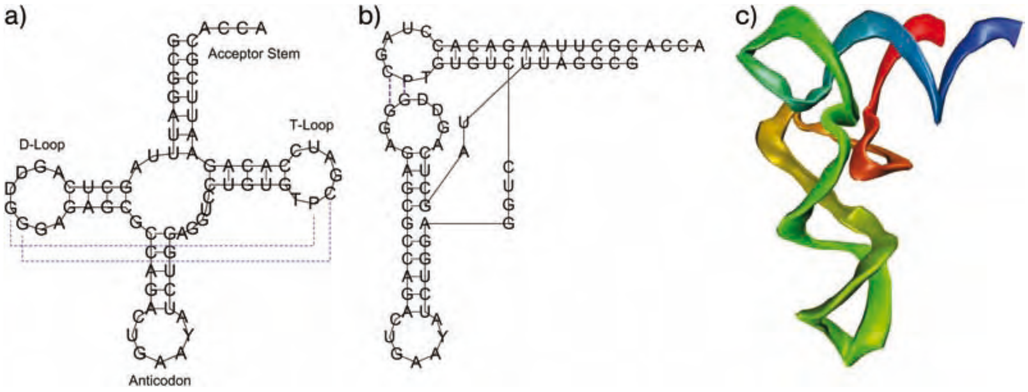


Figure 1 Secondary and tertiary structure of yeast phenylalanine tRNA. (a) The cloverleaf shaped secondary structure consisting of four helices. The dotted blue lines mark evolutionary conserved tertiary contacts. (b) Coaxial stacking results in two extended helices that form the L-shaped tertiary structure. (c) Tertiary structure taken from Protein Data Bank entry 4TRA. The color code (from red to blue) indicates the position along the chain.

prediction and comparison, the focal topics of this chapter, have therefore become a routine tool in the analysis of RNA function.

RNA secondary structures consists of two distinct classes of residues: those that are incorporated in double-helical regions (so called stems) and those that are not part of helices. For RNA, the double-helical regions consist almost exclusively of Watson–Crick C–G and A–U pairs as well as the slightly weaker G–U wobble pairs. All other combinations of pairing nucleotides, called *noncanonical* pairs, are neglected in secondary structure prediction, although they do occur, especially in tertiary structure motifs.

1.2 Secondary Structure Graphs

A secondary structure is primarily a list of base pairs Ω. To ensure that the structure is feasible, a valid secondary structure should fulfill the following constraints:

(i) A base cannot participate in more than one base pair, i.e. Ω is a match on the set of sequence positions.

(ii) Bases that are paired with each other must be separated by at least three (unpaired) bases.

(iii) No two base pairs (i, j) and $(k, l) \in \Omega$ “cross” in the sense that $i < k < j < l$. Matchings that contain no crossing edges are known as loop matchings or circular matchings.

Condition (i) excludes tertiary structure motifs such as base triplets and G-quartets; condition (ii) takes into account that the RNA backbone cannot bend too sharply.

Base pairs that violate condition (iii) are said to form a pseudoknot. While pseudoknots do occur in RNA structures, our definition (somewhat arbitrarily) classifies them as tertiary structure motifs. This is done in part because most dynamic programming algorithms cannot deal with pseudoknots. However, including pseudoknots entails other complications, since most hypothetical structures that violate condition (iii) would also be sterically impossible. Furthermore, little is known about the energetics of pseudoknots, except for some data on H-type pseudoknots [43], the simplest and most common type of pseudoknot (Figure 2). Pseudoknots should therefore be regarded as a first step toward prediction of RNA tertiary structure.

Secondary structures can be represented by "secondary structure graphs" (first two panels in Figure 3). In this representation one creates a graph whose nodes represent nucleotides. There are two kinds of edges: one kind representing the adjacency of nucleotides along the RNA sequence and the other kind representing base pairings. Condition (iii) above assures that this graph is planar, more precisely an *outer-planar graph*, in which all nodes can be arranged along a single face of the planar embedding made up by the edges forming the RNA sequence. We can therefore draw the secondary structure by placing the backbone on a circle and drawing a chord for every base pair such that no two chords intersect.

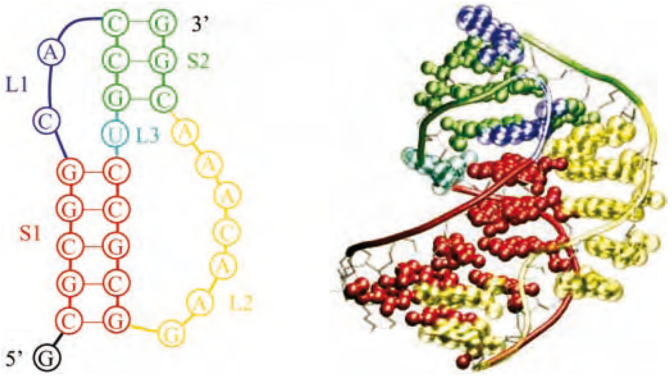


Figure 2 Example of an H-type pseudoknot from beet western yellow virus. The crystal structure (right) shows that the two helices S1 and S2 are coaxially stacked to form a single 3-D helix.

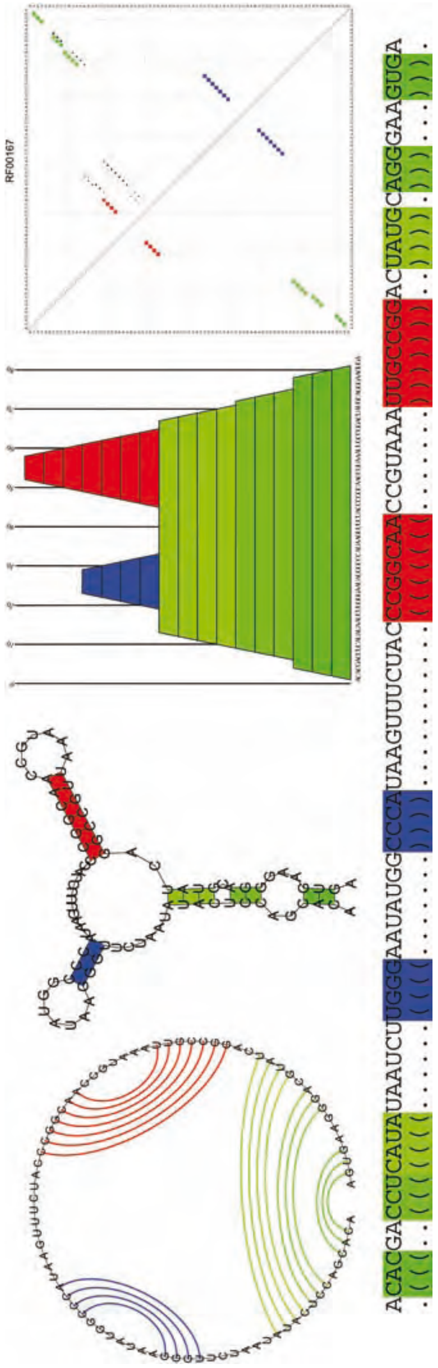

RF00167
ACACGACCUCAUAUAAUCUUGGGAAUAUGGCCCAUAAGUUUCUACCCGGCAACCGUAAAUUGCCGGACUAUGCAGGGAAGUGA

Figure 3 Representations of secondary structures. From left to right: Circle plot, conventional secondary structure graph, mountain plot and dot plot. Removing the backbone edges from the first two representations leaves the matching Ω. Below, the structure is shown in “bracket notation”, where each base pair corresponds to a pair of matching parentheses. The structure shown is the purine riboswitch (Rfam RF00167).

1.3 Mountain Plots and Dot Plots

A representation that works well for large structures and is well suited for comparing structures is the so-called mountain representation. In the mountain representation a single secondary structure is represented in a 2-D graph, in which the x-coordinate is the position k of a nucleotide in the sequence and the y-coordinate the number $m(k)$ of base pairs that enclose nucleotide k (third panel in Figure 3).

Another possible representation is the dot plot, where each base pair (i, j) is represented by a dot or box in row i and column j of a rectangular grid, representing the contact matrix of the structure. Dot plots are well suited to represent structure *ensembles* by superimposing structural possibilities. In particular, they are used to represent thermodynamic ensembles by plotting for each pair a box with area proportional to the equilibrium probability of the pair p_{ij}. Similarly, mfold uses colors to indicate the best possible energy for structures containing a particular pair (right-most panel in Figure 3).

1.4 Trees and Forests

Secondary structures can also be stored compactly in strings consisting of dots and matching brackets: For any pair between positions i and j ($i < j$) we place an open bracket "(" at position i and a closed bracket ")" at j, while unpaired positions in the molecule are represented by a dot (".") (bottom of Figure 3). Since base pairs may not cross, the representation is unambiguous.

An ordered forest F is a sequence of rooted ordered trees $T_1, T_2, \dots, T_m$ such that within each tree T_i the left-to-right order of siblings (children of the same parent) is given. In order to represent RNA secondary structures as ordered forests, we will need to associate a label from a suitable alphabet $\mathcal{A}$ with each node.

This representation of secondary structures in terms of matched parentheses suggests to interpret the structure as a tree [117, 119]. In the *full-tree* representation [32] each base pair corresponds to an interior node and each unpaired base is represented by a leaf (Figure 4). A virtual root vertex is added mostly for graphical reasons.

Leaves may be labeled with the corresponding unpaired base, while interior nodes are labeled with the corresponding base pair. In an extended representation, two leaves, one labeled with the $5'$ and one labeled with the $3'$ nucleotide of the base pair, are attached as the left-most and right-most children to each interior vertex. In this representation the sequence of the molecule can be read of the leaves of the Bielefeld tree in pre-order.

Various coarse-grained representations have been considered. Homeomorphically irreducible trees represent entire helices as interior nodes, while

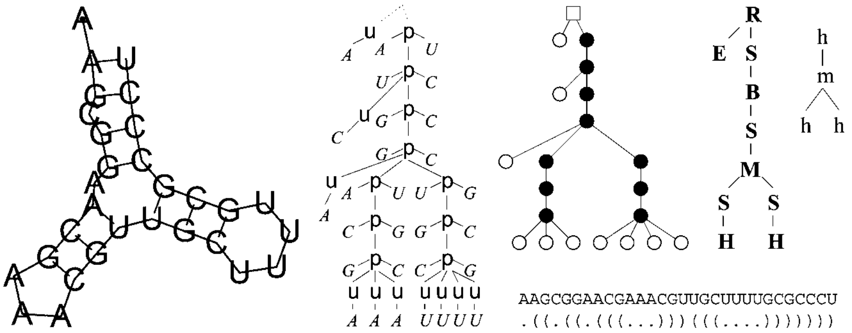


Figure 4 A variety of tree and forest representations of RNA secondary structures have been described in the literature. From left to right: conventional drawing, sequence annotated trees (as e.g. used in RNAforester [51]), "full tree" [32], Shapiro-style tree [117] and branching structure. For comparison, we also show the "bracket notation"

leaves correspond to runs of unpaired bases. Optionally, the length of such a structural element can be used as a weight. Shapiro–Zhang trees [117, 119] explicitly represent the different loop types (hairpin loop, interior loops, bulges, multiloops) as well as stacked regions with special labels. Figure 4 summarizes a few examples.

1.5 Notes

Since RNA secondary structures are planar graphs, they can always be drawn on paper without intersections. Nevertheless, finding a visually pleasing layout is difficult especially for large structures. Layout algorithms for RNA typically make use of the tree-like topology of secondary structure (e.g. Refs. [18, 47, 98, 118]). The problem becomes more complicated when pseudoknots are allowed [46].

2 Loop-based Energy Model

2.1 Loop Decomposition

Secondary structures can be uniquely decomposed into loops, i.e. the faces of the planar drawing of the structure. More formally, we call a position k *immediately interior* of the pair (i, j) if $i < k < j$ and there exists no other base pair (p, q) such that $i < p < k < q < j$. A loop then consists of the closing pair (i, j) and all positions immediately interior of (i, j). As a special case the exterior loop contains all positions not interior of any pair. The loops form a

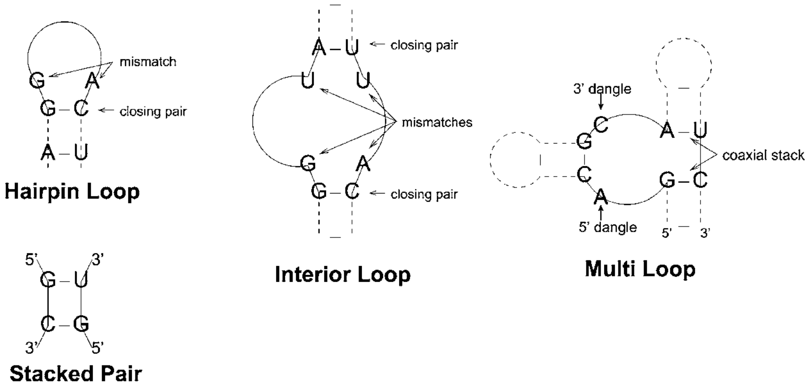


Figure 5 The major types of loops in RNA secondary structures.

minimal cycle basis of the secondary structure graph and this basis is unique for pseudoknot-free structures [80].

A loop is characterized by its length, i.e. the number of unpaired nucleotides in the loop, and its degree, given by the number of base pairs delimiting the loop (including the closing pair). Loops of degree 1 are called hairpin loops, interior loops have degree 2 and loops with degree above 2 are called multiloops (Figure 5). Bulge loops are a special cases of interior loops in which there are unpaired bases only on one side, while stacked pairs correspond to an interior loop of size zero.

The loop decomposition forms the basis of the standard energy model for RNA secondary structures, where the total free energy of a structure is assumed to be a sum over the energies of its constituent loops. As the energy contribution of a base pair in a helix now depends on the preceding and following pair, the model is often called the *nearest-neighbor* model.

2.2 Energy Parameters

Note that a secondary structure corresponds to an ensemble of conformations of the molecule at atomic resolution, i.e. the set of all conformations compatible with a certain base-pairing (hydrogen-bonding) pattern. For example, no information is assumed about the spatial conformation of unpaired regions. The entropic contributions of these restricted conformations have to be taken into account; hence, we are dealing with free energies which will be dependent on external parameters such as temperature and ionic conditions.

Qualitatively, the major energy contributions are base stacking, hydrogen bonds and loop entropies. While hydrogen bond and stacking energies *in vacuo* can be computed using quantum chemistry, the secondary structure model considers free energy differences between folded and unfolded states in an aqueous solution with rather high salt concentrations. As a consequence one has to rely on empirical energy parameters.

Energy parameters are typically derived by following the unfolding of RNA oligomers using a collection of energy parameters is maintained by the group of David Turner [90, 91, 145]. These standard parameters are measured in a buffer of 1 M NaCl at 37°C. As examples we list the free energies for stacked pairs in Table 1. Stacked pairs confer most of the stabilizing energy to a secondary structure, a single additional base pair can stabilize a structure by up to -3.4 kcal mol^{-1}. For comparison, the thermal energy at room temperature is about $RT = 0.6$ kcal mol^{-1}, i.e. the stabilizing energy contribution of a single base pair is typically of the same order of magnitude as the thermal energy. (RNA energy parameters are still published in kcal mol^{-1} to facilitate comparison with previous parameter sets; 1 kcal mol^{-1} $\approx$ 4.2 kJ mol^{-1} in SI units.)

In general, loop energies depend on the loop type and its size. Except for small loops (which are tabulated exhaustively [90]), sequence dependence is conferred only through the base pairs closing the loop and the unpaired bases directly adjacent to the pair. Thus, the loop energy takes the form:

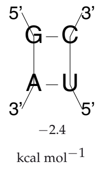
$$E_{\text{loop}} = E_{\text{mismatch}} + E_{\text{size}} + E_{\text{special}}, \tag{1}$$

where E_{mismatch} is the contribution from unpaired bases inside the closing pair and the base pairs immediately interior to the closing pair. The last term is used, for example, to assign bonus energies to unusually stable tetra loops, such as hairpin loops with the sequence motif GNRA. Polymer theory predicts that for large loops E_{size} should grow logarithmically. For multiloops, however, energies that are linear in loop size and loop degree have to be used in order to allow efficient dynamic programming algorithms for structure prediction. While the model allows only Watson–Crick (AU, UA, CG and GC) and wobble pairs (GU, UG), nonstandard base pairs in helices are treated as special types of interior loops in the most recent parameter sets.

The energy model above contains inaccuracies, on the one hand because it assumes that loop energies are strictly additive, on the other hand, because

Table 1 Free energies for stacked pairs (in kcal mol^{-1}).

	CG	GC	GU	UG	AU	UA
CG	−2.4	−3.3	−2.1	−1.4	−2.1	−2.1
GC	−3.3	−3.4	−2.5	−1.5	−2.2	−2.4
GU	−2.1	−2.5	1.3	−0.5	−1.4	−1.3
UG	−1.4	−1.5	−0.5	0.3	−0.6	−1.0
AU	−2.1	−2.2	−1.4	−0.6	−1.1	−0.9
UA	−2.1	−2.4	−1.3	−1.0	−0.9	−1.3



Note that both base pairs have to be read in 5′–3′ direction.

energy parameters carry experimental errors (typically about 0.1 kcal mol^{-1}). Most seriously, the sequence dependence of loop energies has to be kept relatively simple in order to deduce the parameters from a limited number of experiments.

2.3 Notes

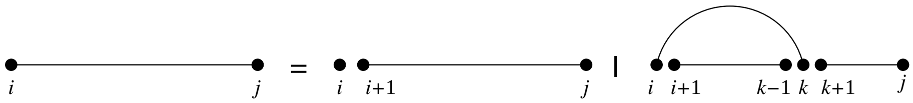
Adjacent helices in multiloops may stack coaxially to form a single extended helix. tRNA structures are prominent examples of this. In the four-armed multiloop the acceptor stem coaxially stacks on the T-stem and the anticodon stem stacks on the D-arm. This results in two extended helices which then form the L-shaped tertiary structure characteristic for tRNAs (Figure 1). Strictly speaking, coaxial stacking goes beyond the secondary structure model, since one has to know *which* helices in the loop will stack in order to include the energetic effect; the list of base pairs is no longer sufficient information to compute the energy. Coaxial stacking is also cumbersome to include in structure prediction algorithms. It has, however, been shown to improve prediction quality [135]. Useful energy parameters for structures with pseudoknots have so far only been collected for simple H-type pseudoknots [43].

3 The Problem of RNA Folding

3.1 Counting Structures and Maximizing Base Pairs

In order to understand the basic ideas behind the dynamic programming algorithms for RNA folding, it is instructive to first consider the underlying combinatorial problem: given an RNA sequence x of length n, enumerate all secondary structures on x. Let x_i denote the i-th position of x. We will simply write "(i, j) pairs" to mean that the nucleotides x_i and x_j *can* form a Watson–Crick or a wobble pair, i.e. $x_i x_j$ is one of GC, CG, AU, UA, GU or UG. A subsequence (substring) will be denoted by $x[i, \ldots, j]$. For notational convenience we interpret $x[j+1, \ldots, j]$ as the empty sequence and associate a single empty structure with it.

The basic idea is that a structure on n nucleotides can be formed in only two distinct ways from shorter structures: either the first nucleotide is unpaired, in which case it is followed by an arbitrary structure on the shorter sequence $x[i+1, \ldots, j]$, or the first nucleotide is paired with some partner base, say k. In the latter case the rule that base pairs must not cross implies that we have independent secondary structures on the subintervals $x[i+1, \ldots, k-1]$ and $x[k+1, \ldots, j]$. Graphically, we can write this decomposition of the set of structures like this:



It is now easy to compute the number N_{ij} of secondary structures on the subsequence $x[i,\ldots,j]$ from positions i to j [139,140]:

$$N_{ij} = N_{i+1,j} + \sum_{k,\,(i,k)\text{ pairs}} N_{i+1,k-1} N_{k+1,j}, \tag{2}$$

with $N_{ii} = 1$. The independence of the structures on $x[i+1,\ldots,k-1]$ and $x[k+1,\ldots,j]$ implies that we can simply multiply their numbers. This simple combinatorial structure of secondary structures was realized by Waterman in the late 1970s [139,140].

Historically, the first attempts at secondary structure prediction tried to maximize the number of base pairs in the structure. The solution to this problem by the Nussinov algorithm [101] is very similar to the combinatorial recursion above. Denote by E_{ij} the maximal number of base pairs in a secondary structure on $x[i,\ldots,j]$. Using the decomposition of the structure set, we see that E_{ij} is the optimal choice among each of the alternatives. In this context, independence of two substructures in the paired cases implies that we have to optimize these substructures independently. If we like, we can associate each base pair with a weight (negative energy) β_{ij} which depends on x_i and x_j; we arrive immediately at the recursion:

$$E_{ij} = \max\left\{ E_{i+1,j}, \max_{k,\,(i,k)\text{ pairs}} \left\{ E_{i+1,k-1} + E_{k+1,j} + \beta_{ik} \right\} \right\}. \tag{3}$$

Replacing the weights by binding energies (which are negative for stabilizing interactions) we simply have to replace max by min in the above recursions. In practice, this simplified energy model does not lead to reasonable predictions in most cases. We use it here for didactic purposes and relegate a more detailed description of the complete RNA folding problem to Section 3.3.

The energy contributions of individual base pairs are of the same order of magnitude as the thermal energy at room temperature. Thus, RNA molecules exist in a distribution of structures rather than in a single ground-state structure. Thermodynamics dictates that, in equilibrium, the probability of a particular structure Ψ is proportional to its Boltzmann factor $\exp[-E(\Psi)/RT]$. Here $E(\Psi)$ is the energy of the sequence in conformation (secondary structure) Ψ, R is the molar gas constant (Boltzmann's constant in molar units) and T is the absolute ambient temperature in Kelvin. This ensemble of structures is determined by its *partition function*:

$$Z = \sum_{\Psi} \exp(-E(\Psi)/RT)\,, \tag{4}$$

or, equivalently, by the free energy $\Delta G = -RT \ln Z$. The partition function Z can be computed in analogy to Eq. (3). Using Z_{ij} as the partition function over all structures on subsequence $x[i,\ldots,j]$ we obtain [93]:

$$Z_{ij} = Z_{i+1,j} + \sum_{k,\,(i,k)\text{ pairs}} Z_{i+1,k-1} Z_{k+1,j} \exp(-\beta_{ik}/RT)\,. \tag{5}$$

Note that we can transform the recursion for E_{ij} in Eq. (3) into the equation for Z_{ij} simply by exchanging maximum operations with sums, sums with multiplications and energies by their corresponding Boltzmann factors.

The partition function allows us to compute the equilibrium probability of a structure Ψ as $p(\Psi) = \exp[-E(\Psi)/RT]/Z$. The formalism is also used to efficiently compute the equilibrium probability of a base pair $p_{ij} = \sum_{(i,j)\in\Psi} p(\Psi)$. To this end one needs to compute the partition function $\widehat{Z}_{ij}$ of structures *outside* the subsequence $x[i,\ldots,j]$ using a recursion similar to the one above for Z. We can now compute the partition function over all structures containing the pair (i,j) and thus its probability:

$$p_{ij} = \widehat{Z}_{ij} Z_{i+1,j-1} \exp(-\beta_{ij}/RT)/Z\,. \tag{6}$$

Further variants of this scheme can be employed to compute, for example, the number of states with a given energy, to explicitly list all possible structures or to determine structures that optimize other properties. In Section 5.5 we will briefly mention how such variants can be constructed in a systematic way within the framework of *algebraic dynamic programming* (ADP) [36].

3.2 Backtracing

Recursion (3) computes only the optimal energy, not an optimal structure which realizes this energy. This is typical for most dynamic programming algorithms: one first computes the value of the optimum, then uses *backtracing* (sometimes called *backtracking*) to generate one (or more) structures in a stepwise fashion based on the information collected in the forward recursions. This section closely follows an exposition of the topic in Ref. [29]. The basic object is a partial structure π consisting of a collection Ω_π of base pairs and a collection Υ_π of sequence intervals in which the structure is not (yet) known. Positions that are known to be unpaired can easily be inferred from this information. The completely unknown structure on the sequence interval $[1,n]$ is therefore $\varnothing = (\varnothing, \{[1,n]\})$ while a structure is complete if it is of the form $\pi = (\Omega, \varnothing)$.

Suppose $I = [i, j] \in \Upsilon$ are positions for which the partial structure $\pi = (\Omega, \Upsilon)$ is still unknown. If we know that i is unpaired, then $\pi' = (\Omega', \Upsilon')$ with $\Omega' = \Omega\ \Upsilon' = \Upsilon \setminus \{I\} \cup \{[i+1, j]\}$. If $(i, k), i < k \leq j$, is a base pair, then $\Omega' = \Omega \cup \{(i, k)\}$ and $\Upsilon' = \Upsilon \setminus \{I\} \cup \{[i+1, k-1], [k+1, j]\}$. Here we use the convention that empty intervals are ignored. Furthermore, base pairs can only be inserted within a single interval of the list Υ. We write $\pi' = \pi \blacktriangleleft (i)$ and $\pi' = \pi \blacktriangleleft (i, k)$ for these two cases.

The energy of a partial structure π is defined as:

$$E(\pi) = \sum_{(k,l) \in \Omega} \beta_{kl} + \sum_{I \in \Upsilon} E_{\text{opt}}(I)\,, \tag{7}$$

where $E_{\text{opt}}(I) = E_{ij}$ is the optimal energy for the substructure on the interval $I = [i, j]$.

The standard backtracing for the minimal energy folding starts with the unknown structure. Instead of a recursive version we describe here a variant where incomplete structures are kept on a stack $\mathfrak{S}$. We write $\pi \leftarrow \mathfrak{S}$ to mean that π is popped from the stack and $\pi \rightarrow \mathfrak{S}$ to mean that π is pushed onto the stack.

If we want all optimal energy structures instead of a single representative we simply test all alternatives, i.e. we omit the **next** in Algorithm B1, Table 2. It is now almost trivial to modify the backtracing to produce all structures within an energy band $E_{\text{opt}} \leq E \leq E_{\text{max}}$ above the ground state.

Stochastic backtracing procedures for dynamic programming algorithm such as pairwise sequence alignment are well known [97]. Replacing Z_{ij} by N_{ij} in Algorithm B3 we recover recursions for producing a uniform ensemble of structures similar to the procedure for producing random structures without sequence constraint used in Ref. [127].

Note that the probabilities of $\pi \blacktriangleleft (i+1)$ and $\pi \blacktriangleleft (i, k)$ for all k add to 1 so that in each iteration we take exactly one step. Hence, we simply fill one structure which we output as soon as it is complete. See Table 2.

3.3 Energy Minimization in the Loop-based Energy Model

Using the loop-based energy model is essential in order to achieve reasonable prediction accuracies. As we shall see, the more complicated energy model results in somewhat more complicated recursions and requires additional tables. However, memory and CPU requirements are still $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$. The main difference from the simple model discussed in the previous sections is that we now have to distinguish between different types of loops. Thus, we have to further decompose the set of substructures enclosed by the base pair (i, k) according to the loop types: hairpin loop, interior loop

Table 2 Comparison of backtracing recursions for different algorithms

Algorithm B1 [101, 150]: Backtracing a single structure	**Algorithm B2** [144]: Backtracing multiple structures	**Algorithm B3** [21]: Stochastic backtracing

$\varnothing \to \mathfrak{S}$.
while$\mathfrak{S} \neq \varnothing$
$\pi \leftarrow \mathfrak{S}$;
if π is complete **then** output π
$[i,j] = I \in \Upsilon_\pi$.
$\pi' = \pi \blacktriangleleft (i+1)$
if $E(\pi') = E_{\text{opt}}$ **then** $\pi' \to \mathfrak{S}$;
next;
for all$k \in [i,j]$ **do**
$\pi' = \pi \blacktriangleleft (i,k)$
if $E(\pi') \leq E_{\text{opt}}$
then $\pi' \to \mathfrak{S}$; **next**;
	$\varnothing \to \mathfrak{S}$.
	while $\mathfrak{S} \neq \varnothing$
	$\pi \leftarrow \mathfrak{S}$;
	if π is complete **then** output π
	for all $[i,j] = I \in \Upsilon_\pi$ **do**
	$\pi' = \pi \blacktriangleleft (i+1)$
	if $E(\pi') \leq E_{\text{opt}} + \Delta E$
	then $\pi' \to \mathfrak{S}$;
	for all$k \in [i,j]$ **do**
	$\pi' = \pi \blacktriangleleft (i,k)$
	if $E(\pi') \leq E_{\text{opt}} + \Delta E$
	then $\pi' \to \mathfrak{S}$;
		$\varnothing \to \mathfrak{S}$.
		while $\mathfrak{S} \neq \varnothing$
		$\pi \leftarrow \mathfrak{S}$;
		if π is complete **then** output π
		for all $[i,j] = I \in \Upsilon_\pi$ **do**
		$\pi' = \pi \blacktriangleleft (i+1)$
		$\pi' \to \mathfrak{S}$ with probability
		$Z(\pi')/Z(\pi)$
		for all$k \in [i,j]$ **do**
		$\pi' = \pi \blacktriangleleft (i,k)$
		$\pi' \to \mathfrak{S}$
		with prob. $Z(\pi')/Z(\pi)$

and multi(branched) loops (Figure 6). The hairpin and interior loop cases are simple since they reduce again to the same decomposition step.

The multiloop case is more complicated, however, since the multiloop energy depends explicitly on the number of substructures ("components") that emanate from the loop. We therefore need to decompose the structures within the multiloop in such a way that we can at least implicitly keep track of the number of components. To this end we represent a substructure within a multiloop as a concatenation of two components: an arbitrary 5′ part that contains *at least* one component, and a 3′ part that starts with a base pair and contains only a single component. These two types of multiloop substructures are now decomposed further into parts that we already know: unpaired inter-

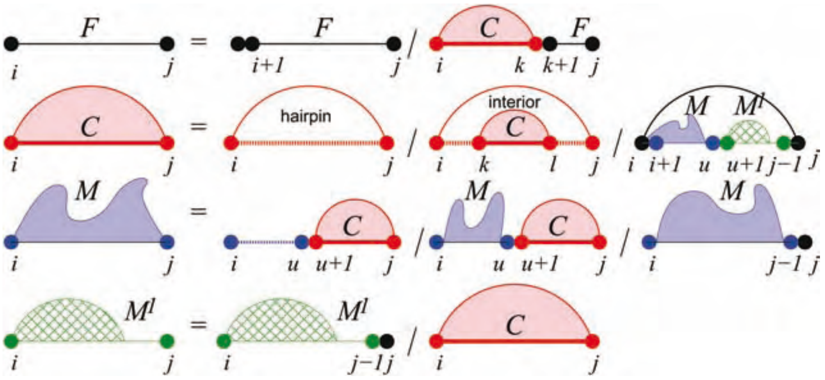


Figure 6 Decomposition of RNA secondary structure. Dotted lines indicate unpaired substructures, while full lines denote arbitrary structures; base pairs are indicated as arcs. Multiloop contributions with an arbitrary number of components are shown as irregular "mountains". See text for further details.

vals, structures enclosed by a base pair and (shorter) multiloop substructures (Figure 6). It is not too hard to check that this decomposition really accounts for all possible structures and that each secondary structure has a unique decomposition.

Given the recursive decomposition of the structures, we can now rather easily derive the associated energy minimization algorithm. We will use the abbreviations $\mathcal{H}(i,j)$ for the energy of a hairpin loop closed by the pair (i,j), similarly $\mathcal{I}(i,j;k,l)$ shall denote the energy of an interior loop determined by the two base pairs (i,j) and (k,l). We will also tabulate the following quantities:

F_{ij} free energy of the optimal substructure on the subsequence $x[i,\dots,j]$.

C_{ij} free energy of the optimal substructure on the subsequence $x[i,\dots,j]$ subject to the constraint that i and j form a base pair.

M_{ij} free energy of the optimal substructure on the subsequence $x[i,\dots,j]$ subject to the constraint that that $x[i,\dots,j]$ is part of a multiloop and has at least one component.

M^1_{ij} free energy of the optimal substructure on the subsequence $x[i,\dots,j]$ subject to the constraint that that $x[i,\dots,j]$ is part of a multiloop and has exactly one component, which has the closing pair i,h for some h satisfying $i < h \leq j$.

The recursions for computing the minimum free energy of an RNA molecule in the loop based energy model were first formulated by Zuker and Stiegler [150]. They can be summarized as follows:

$$\begin{aligned}
F_{ij} &= \min\left\{F_{i+1,j}, \min_{i<k\leq j}\left(C_{ik}+F_{k+1,j}\right)\right\} \\
C_{ij} &= \min\left\{\begin{matrix}\mathcal{H}(i,j), \min_{i<k<l<j}\left(C_{kl}+\mathcal{I}(i,j;k,l)\right), \\ \min_{i<u<j}\left(M_{i+1,u}+M^1_{u+1,j-1}+a\right)\end{matrix}\right\} \\
M_{ij} &= \min\left\{\begin{matrix}\min_{i<u<j}\left((u-i+1)c+C_{u+1,j}+b\right), \\ \min_{i<u<j}\left(M_{i,u}+C_{u+1,j}+b\right),\ M_{i,j-1}+c\end{matrix}\right\} \\
M^1_{ij} &= \min\left\{M^1_{i,j-1}+c,\ C_{ij}+b\right\},
\end{aligned} \tag{8}$$

where we assume linear multiloop energies of the form $E_{\mathrm{ML}} = a + b \cdot \text{degree} + c \cdot \text{size}$. In contrast to most implementations the version shown here decomposes structures in such a way that each substructure occurs exactly once.

While this is not strictly necessary for energy minimization, it allows us to use essentially the same recursions for all variants of the problem, including the computation of the partition function, or the backtracing of all or a sample of suboptimal structures.

3.4 RNA Hybridization

Intermolecular base pairing between two RNA molecules can be treated in the same way as intramolecular interactions. The most straightforward approach is to concatenate the two molecules. One can then apply the folding algorithms for single molecules. There is only a single necessary modification to the folding algorithms: the energy contribution of the loop that contains the cut point is different. Implementations of this approach are RNAcofold [56] and pairfold [5].

From a physics point of view, however, additional effects need to be taken into account: the interaction of two distinct molecules is concentration dependent. Furthermore, there is an additional (entropic) contribution for the initiation of an intermolecular interaction. The extension of the folding algorithms of course compute both inter- and intra-molecular contributions. It is therefore necessary to correct for the initialization energy E^i:

$$\begin{aligned} Z_{AA} &= (Z^\circ_{AA} - Z^2_A)\exp(-E^i/RT) \\ Z_{BB} &= (Z^\circ_{BB} - Z^2_B)\exp(-E^i/RT) \quad \text{and} \\ Z_{AB} &= (Z^\circ_{AB} - Z_A Z_B)\exp(-E^i/RT)\,. \end{aligned} \tag{9}$$

where Z° is the partition function as calculated from the folding algorithm for the concatenated sequences, and Z_A and Z_B are the partition functions of the isolated molecules A and B. Standard statistical thermodynamics can then be used to compute the concentration dependencies of the complex formation (e.g. Ref. [20] for a discussion in the context of RNA hybridization).

Various simplified approaches have been discussed in recent years. In particular, the most common approximation is to neglect the secondary structures of the two interacting molecules. This amounts to a model in which the concatenated structure can only have base pairs and interior loops, and the cut point is located in the single hairpin loop. It does, however, result in a much faster algorithm with time complexity $\mathcal{O}(n \cdot m)$ instead of $\mathcal{O}((n+m)^3)$ for two sequences of length n and m. Algorithms for this case have been described in Refs. [20, 105]; the Vienna RNA Package also provides an implementation. The RNAhybrid program was in particular used to detect microRNA/target interactions.

3.5 Pseudoknotted Structures

Many functionally important RNA structures contain pseudoknots, including rRNAs [12], RNase P RNAs [10,49] and tmRNA [151]. Recently, algorithms have been described that are able to deal with certain classes of pseudoknotted structures. However, as we shall see below, these are plagued by considerable computational costs. In addition, a common problem of all these approaches is the still very limited information about the energetics of pseudoknots [43,66].

In the general case of unrestricted pseudoknots the problem is NP-complete when a loop-based energy model is used [3,86]. Arbitrarily complex pseudoknots, however, are also biologically unrealistic. While every secondary structure has a plausible 3-D realization (this follows directly from the tree structure of secondary structures), this is not true for more general structures: it may well be impossible to embed a given arbitrary set of base pairs in 3-D space such that chemically reasonable distances are maintained. By construction, such constraints cannot be incorporated into our graph-based model of RNA structure. One remedy is to restrict oneself to certain (simple) classes of pseudoknots.

Figure 7 shows the algorithmic problem with pseudoknots. In principle, one could include (a certain type of) pseudoknots as additional structural elements into the dynamic programming recursion. The further decomposition of the structure, however, requires at least two coupled cut points, here k and l, even if we assume that the structures of crossing arc sets are only single stems. This increases the CPU requirements to at least $\mathcal{O}(n^4)$. More realistic models, such as the H-type model, require additional memory as well.

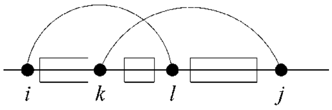


Figure 7 Additional requirements for computing pseudoknotted structures. In order to evaluate the contribution for the pseudoknot on $[i, j]$ we need to iterate over all combinations of cutpoints $k < l$.

Algorithms for a number of different classes of pseudoknots have been published in recent years, (e.g. Refs. [3,22,86,104,109]). Figure 8 summarizes the relationships between the algorithmic complexities of predicting secondary structures from some of these structure classes [16].

3.6 Notes

The basic counting recursion can be readily modified to enumerate other quantities of interest such as the structures with particular properties and distributions of structural elements (e.g. Ref. [58]). The combinatorics of RNA

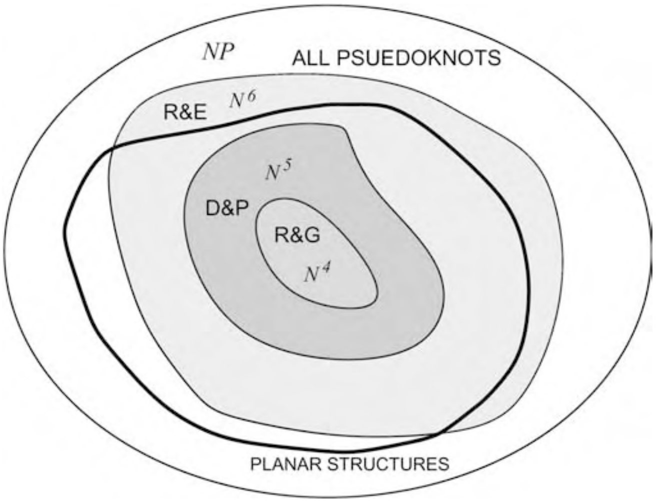


Figure 8 The most prominent classes of pseudoknotted structure are those investigated by Reeder and Giegerich R&G [104], Dirks and Pierce D&P [22], and Rivas and Eddy R&E [109].

secondary structures and related mathematical objects such as ordered trees, Motzkin paths and noncrossing partitions is still an active area of research (e.g. Refs. [13, 19, 81] and the references therein).

The recursions for the loop-based energy model as displayed above, in fact, give rise to $\mathcal{O}(n^4)$ CPU requirements due to the interior loop contribution. However, very long interior loops are extremely unlikely (and unstable), so that the length of interior can be bounded by a constant, e.g. $M = 30$. The interior loop contribution thus remains quadratic. Under certain plausible assumptions on the interior loop energies, a cubic time algorithm can be designed [87] that takes interior loops of all sizes into account.

A restriction of the folding algorithm to local structure is described in Ref. [57]. Here, the maximum span $|j - i + 1|$ of a base pair (i, j) is bounded by a constant L. The resulting "scanning" algorithms are linear in time and space, and hence can be used to screen entire genomes for locally stable structures.

Circular RNA molecules are rare, but their secondary structures are of considerable interest because structural features are important, e.g. in viroids [108,123]. A straightforward way of dealing with circular RNA molecules is to compute C_{ij} and M_{ij} also for the subsequences of the form $x[j, \dots, n]x[1, \dots, i]$ [149]. The disadvantage of this approach is, however, that it doubles the memory requirements. An alternative is described in Ref. [60].

A secondary structure Ω is *saturated* if none of its stems can be elongated, i.e. if any single base pair that is inserted into Ω does not stabilize the structure by stacking to any other base pairs. The recursions (Figure 6) can be modified to produce only saturated structures [26]. Similarly, we may call Ω *locally base pair optimal* if Ω cannot be expanded by any additional base pair. In Ref. [15] a dynamic programming algorithm is described that computes such locally optimal structures in quartic time with cubic memory requirements.

Prediction of pseudoknotted structures based on maximum matching can be done using algorithms for Maximum Weighted Matching [125]. While this approach requires only $O(n^3)$ time, it cannot take the loop-based energy model into account. "Iterated loop matching", i.e. the repeated (greedy) application of the Nussinov algorithm, is another approximate way of computing pseudoknotted structures [113]. Finally, heuristics such as genetic algorithms can be used to compute pseudoknotted structures [78].

4 Conserved Structures, Consensus Structures and RNA Gene Finding

4.1 The Phylogenetic Method

Most functional RNA molecules have characteristic secondary structures that are highly conserved in evolution. Well-known examples include rRNAs, tRNAs, RNase P and MRP RNAs, the RNA component of signal recognition particles, tmRNA, group I and group II introns, and small nucleolar RNAs. It is therefore of considerable practical interest to efficiently compute the consensus structure of a collection of such RNA molecules.

Given a sufficiently large database of aligned RNA sequences, one can directly infer a consensus secondary structure from the data. The basic idea is that substitutions in the sequence will respect the common structural constraints. Therefore, substitutions in helical regions have to be correlated, since in general only six (GC, CG, AU, UA, UG and GU) out of the 16 combinations of two bases can be incorporated in the helix. Two columns in the alignment thus will covary if they form a base pair.

For concreteness, assume that we are given a multiple sequence alignment $\mathbb{A}$ of N sequences. By $\mathbb{A}_i$ we denote the i-th column of the alignment, while a_i^α is the entry in the α-th row of the i-th column. The length of $\mathbb{A}$, i.e. the number of columns, is n. Furthermore, let $f_i(\mathrm{X})$ be the frequency of base X at aligned position i and let $f_{ij}(\mathrm{XY})$ be the frequency of finding simultaneously X at position i and Y at j.

The most common way of quantifying sequence covariation for the purpose of RNA secondary determination is the *mutual information* score [14,44,45]:

$$MI_{ij} = \sum_{\mathrm{X,Y}} f_{ij}(\mathrm{XY}) \log \frac{f_{ij}(\mathrm{XY})}{f_i(\mathrm{X}) f_j(\mathrm{Y})}. \tag{10}$$

Usually, the mutual information score makes no use of RNA base-pairing rules. For large datasets this is desirable, since it allows us to identify noncanonical base pairs and tertiary interaction. For the small datasets considered in the following subsections, however, neglecting base pairing rules does more harm (by increasing noise) than good. In particular, mutual information does not account at all for consistent noncompensatory mutations, i.e. if we have, say, only GC and GU pairs at positions i and j then $M_{ij} = 0$. Thus, sites with two different types of base pairs are treated just like a pair of conserved positions.

A straightforward measure of covariation takes the form:

$$C_{ij} = \sum_{\mathrm{XY},\mathrm{X'Y'}} f_{ij}(\mathrm{XY}) \mathbf{D}_{\mathrm{XY},\mathrm{X'Y'}} f_{ij}(\mathrm{X'Y'}). \tag{11}$$

where a suitable choice for the 16×16 matrix $\mathbf{D}$ has entries $\mathbf{D}_{\mathrm{XY},\mathrm{X'Y'}} = d_H(\mathrm{XY},\mathrm{X'Y'})$ if both $\mathrm{XY} \in \mathcal{B}$ and $\mathrm{X'Y'} \in \mathcal{B}$ and $\mathbf{D}_{\mathrm{XY},\mathrm{X'Y'}} = 0$ otherwise. Here $\mathcal{B} = \mathrm{GC}, \mathrm{CG}, \mathrm{AU}, \mathrm{UA}, \mathrm{GU}$ or UG, and $d_H(\mathrm{XY},\mathrm{X'Y'})$ is the Hamming distance of XY and X′Y′. The idea here is that consistent mutations such as GC → GU should count less (here half) of a compensatory mutation such as GC → AU. Note that Eq. (11) is a scalar product, $C_{ij} = \langle f_{ij} \mathbf{D} f_{ij} \rangle$ and hence can be evaluated efficiently. If desired, $\mathbf{D}$ could be replaced by a different kernel that, for example, could incorporate measured substitution rates [35].

The purely phylogenetic approach suffers from two limitations. (i) It requires a very large set of sequences in order to obtain a reliable estimate of covariance or mutual information for each pair of sequences. With the exception of rRNAs and tRNAs, such large datasets are usually not (yet) available. (ii) It is sensitive to alignment errors and hence not applicable to very diverse sets of sequences. A possibly remedy is provided by approaches towards solving the folding and alignment problems simultaneously or iteratively. These are discussed in the following section.

4.2 Conserved Structures

The amount of data that is required for inferring structures can be reduced dramatically by taking thermodynamics of folding into account. Indeed, Ref. [48] suggested to resolve ambiguities in the phylogenetic analysis based on thermodynamic considerations.

However, the converse approach, i.e. to use the information which base pairs are thermodynamically plausible, appears to be more efficient. Most of the alignment-based methods therefore start from thermodynamics-based folding and use the analysis of sequence covariations or mutual information for postprocessing (see, e.g. Refs. [54, 59, 71, 77, 84, 85]). We describe here the alidot algorithm (Figure 9) [54, 59].

For each of the aligned sequences secondary structures are computed separately. The resulting lists of base pairs from either minimum free energy calculations or from a partition function calculation are then superimposed by using the multiple sequence alignment to determine which pairs in different sequences are equivalent. For each pair we now have both thermodynamic and sequence covariation information, which is used to hierarchically rank order base pairs depending on their support across the entire data set. A greedy procedure then extracts contiguous stems from the rank ordered list and combines them to a partial secondary structure which contains only those sequence/structure elements that are significantly conserved throughout the aligned input sequences.

An alternative to the ranking/greedy approach of alidot is to compute a score or weight w_{ij} for each possible base pair. The program ConStruct [84] uses a simple scoring function that exclusively combines the base-pairing probabilities of the individual sequences. Covariance or mutual information score as well as contributions that consider the potential to extend the pair to a longer helix [141] could easily be included. A secondary structure can then be computed using the Nussinov algorithm with weights w_{ij} for the base pairs. The downside of this approach is that it returns a global secondary structure rather than a collection of well-supported local features.

Comparative approaches are based on the fact that RNA secondary structure is quite fragile against randomly placed point mutations. Our earlier computational studies suggest that even with 85% sequence identity we should expect no significant structural similarity [33, 116]. While this result

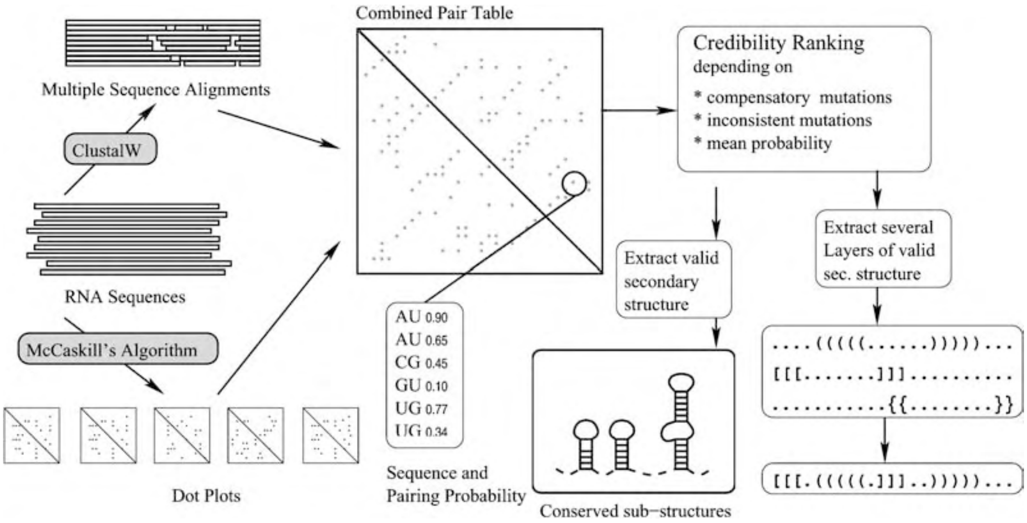


Figure 9 Flow chart of the alidot algorithm.

may seem surprising, there has been convincing experimental evidence (see e.g. Ref. [115]). Methods such as alidot thus can discriminate very well between conserved and nonconserved RNAs. Both alidot and ConStruct require interactive work and are therefore best suited for small genomes as found in RNA viruses [61,131,142].

4.3 Consensus Structures

Sometimes it is known *a priori* that the aligned sequences should fold into a common secondary structure. This is the case, for example, for rRNAs, tRNAs and many other small noncoding RNA (ncRNA) molecules. In this case it makes sense to ask, what is the most stable structure that can be formed simultaneously by all (or almost all) input sequences? This problem is solved in a rather straightforward way by RNAalifold [55]. It treats the entire alignment like a single sequence and solves the secondary structure problem for this "generalized sequence". To this end, of course, an extension of the standard energy model to alignments is required. RNAalifold simply averages the energy contribution over all sequences. In the simple case of base-pair-dependent energies this means:

$$\beta_{ij}^{\mathbb{A}} = \frac{1}{N} \sum_{\alpha} \beta_{x_i^\alpha, x_j^\alpha}. \tag{12}$$

For the realistic energy model, energies for the different loop types are averaged individually.

Both the mutual information score and the covariance score assign a bonus to compensatory mutation. Neither score deals with inconsistent sequences, i.e. with sequences that cannot form a base pair between positions i, j. The simplest ansatz for this purpose is to simply count the number of sequences q_{ij} that cannot form a canonical base pair between columns i and j. Here, combinations of a nucleotide and a gap are counted as inconsistent while gap–gap combinations (i.e. deletions of an entire base pair) are ignored.

In a multiple alignment of a larger number of sequences we have to expect occasional sequencing errors and of course there will be alignment errors. Thus, we cannot simply mark a pair of positions as nonpairing if a single sequence is inconsistent. Furthermore, there is the possibility of a nonstandard base pair [44]. Thus, we define a threshold value for the combined score $B_{ij} = C_{ij} - \phi_1 q_{ij}$ and declare a pair of positions i, j as nonpairing if B_{ij} is too small.

Figure 10 shows the consensus structure of the mir-105 microRNA family as an example. Such consensus structures are needed for the derivation of pattern descriptions that can be used to search for structurally similar RNAs in genomic DNA, as briefly described in the following section.

4.4 RNA Gene Finding

It is, of course, possible to identify genomic sequences that are homologous to known RNA genes, using either BLASTN or, as in the case of tRNAs, more specialized methods. For most functional ncRNA molecules the secondary

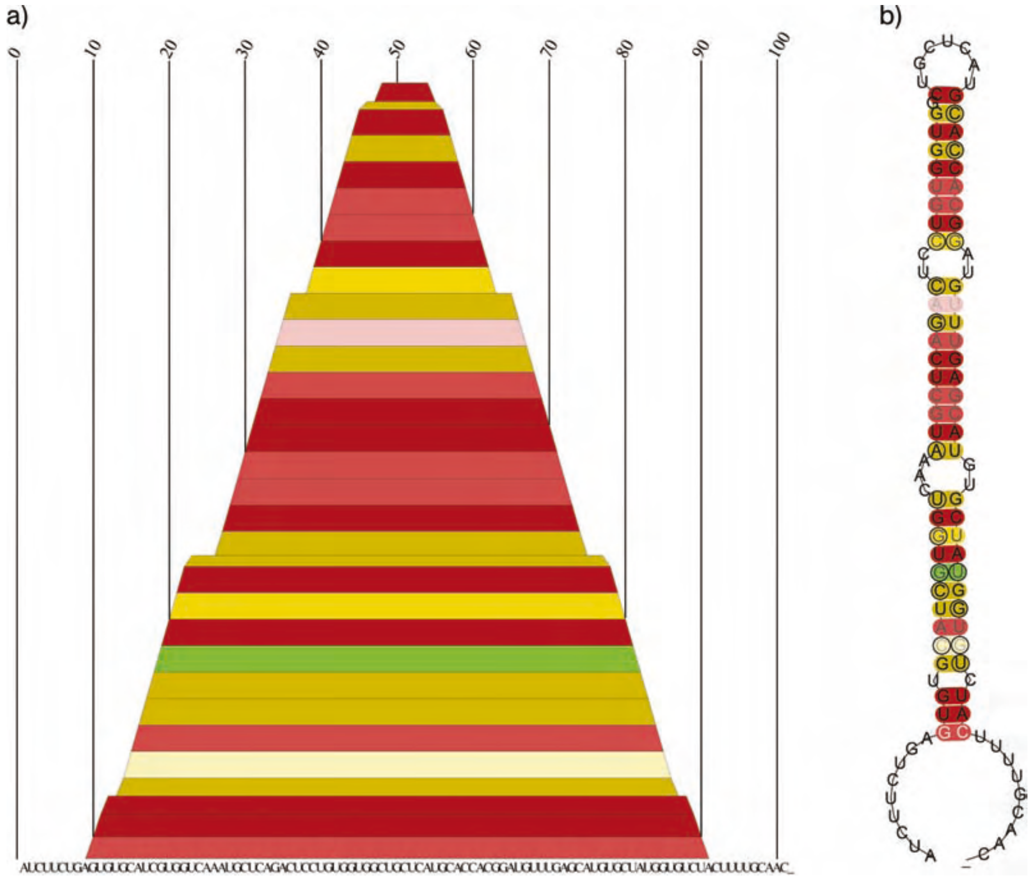

a)
0
10
20
30
40
50
60
70
80
90
100
AUCUUCUGAGUGUGCAUCGUGGUCAAAUGCUCAGACUCCUGUGGUGGCUGCUCAUGCACCACGGAUGUUUGAGCAUGUGCUAUGGUGUCUACUUUUGCAAC_
b)

Figure 10 Consensus secondary structure of the 11 sequences from mammalian microRNA mir-105. Sequences are taken from the microRNA Registry (version 6.0) and from BLAST searches in vertebrate genomes. (a) *Mountain plot:* a base pair (i, j) is represented by a slab ranging from i to j. The $5'$ and $3'$ sides of stems thus appear as up-hill and down-hill slopes, respectively, while plateaus indicate unpaired regions. Colors indicate sequence variation by encoding the number of different types of base pairs (GC, CG, AU, UA,GU, UG) that occur in the two paired columns of the alignment. Pairs with conserved sequence are shown in red; ocher, green, cyan, blue and violet indicate two to six types of base pairs. Pairs with one or two inconsistent mutations are shown in (two degrees of) pale colors. (b) In the *conventional secondary structure graph* paired positions are color coded as in the mountain plot. Consistent mutations are indicated by circles around the varying position, compensatory mutations thus are marked by circles around both pairing partners.

structure is much more conserved than their sequence. This can be used to identify putative ncRNA sequences using programs such as RNAmot [34], tRNAscan [83] or HyPa [40]. Nevertheless, all these approaches are restricted to searching for new members of the few well-established families such as tRNAs, small nucleolar RNAs, microRNAs and certain spliceosomal RNAs.

A different approach is taken in the program QRNA [110]. This method for comparative analysis of two aligned homologous sequences can detect novel structural RNA genes by deciding whether the substitution pattern fits better with (i) synonymous substitutions, which are expected in protein-coding regions, (ii) the compensatory mutations consistent with some base-paired secondary structure or (iii) uncorrelated mutations.

The alidot approach has never been used for large-scale gene finding since it has turned out to be nontrivial to assign statistical significance values to its results. Most recently, however, a conceptually related technique has been developed that is efficient and sensitive enough to allow genome-wide screens for RNAs.

The program RNAz [138] combines a comparative approach (scoring conservation of secondary structure) with the observation [8,76,136] that ncRNAs are thermodynamically more stable than expected by chance. This excess stability is conveniently measured in terms of the z-score:

$$z = \frac{E - \overline{E}}{\sigma}, \tag{13}$$

where $\overline{E}$ and σ are mean and standard deviation of the distribution of shuffled sequences. Instead of dealing with individual sequences, RNAz uses multiple sequence alignments of potential RNAs from different species as input. The computation of z by direct sampling is extremely time-consuming. In RNAz it is therefore replaced by a support vector machine that has been trained to solve the regression problems of estimating $\overline{E}$ and σ from properties of the input sequences.

Structural conservation is also quantified in thermodynamical terms. The structure conservation index S is defined as the ratio of the average energy of the consensus structure (as computed by RNAalifold) and the average of the unconstrained folding energies of the individual sequences. An alignment of identical sequences thus has $S = 1$. On the other hand, completely unrelated sequences will not be able to form a consensus structure since there are always some sequences that contradict any particular pairing, thus $S = 0$. Sequences that form a well-conserved consensus sequence in the presence of sequence covariations, finally, will have the same energy contributions in the consensus and in the individual folds. In addition, however, the consensus energy contains the bonus contributions for sequence covariations, so that we obtain $S > 1$. See Figure 11.

RNAz uses a support vector machine (SVM) [17] to determine from the z-score and the structure conservation index whether a given multiple sequence alignment is a structurally conserved RNA. Surveys of animal genomes [96,

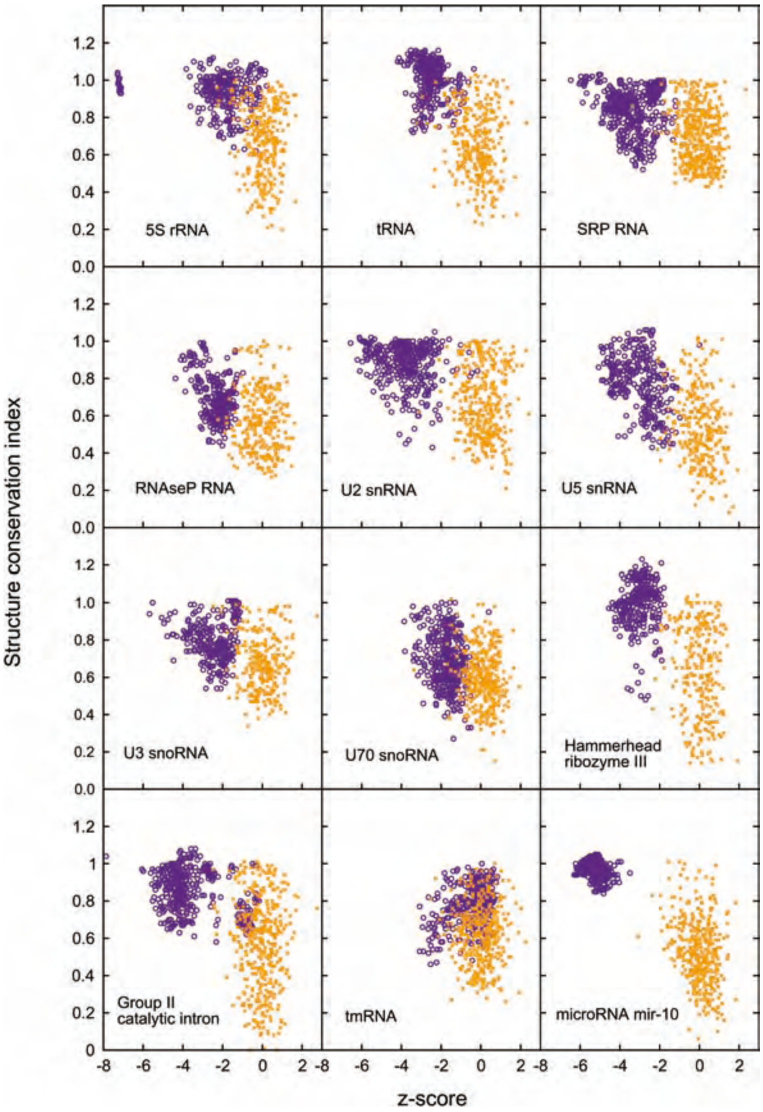


Figure 11 Scatter plot of structure conservation index S (x-axis) and energy z-score (y-axis) for different families of structured ncRNAs. In each panel, the properties of the true sequences (dark) are compared with controls obtained by shuffling the sequence. Data are taken from Ref. [138].

137] reveal a very large number of previously unknown candidates for both independent ncRNAs and structured *cis*-acting elements in mRNAs.

4.5 Notes

Including covariation information is also a good way to improve the accuracy of structure predictions including pseudoknots. One approach is to forgo a loop-based energy model and use base pair scores instead, in which case the resulting Maximum Weighted Matching problem can be solved efficiently [125]. Good accuracies can be achieved by using a combination of covariance and thermodynamic criteria for scoring potential base pairs [141]. The ILM program of Ruan and coworkers [113], uses the Nussinov algorithm iteratively in order to build pseudoknotted structures.

5 Grammars for RNA Structures

5.1 Context-free Grammars (CFGs) and RNA Secondary Structures

The recursions for RNA folding in Figure 6 suggest a close connection with certain grammars. More precisely, we may interpret Figure 6 as the production rules of an "RNA language". The tree representations in Figure 4, on the other hand, are suggestive of a connection between RNA structures and parse trees of a grammar that generates RNA sequence. As we shall see in this section, these connections can be made precise and open the door to the application of learning techniques in RNA bioinformatics.

Recall that a formal language $\mathcal{L}$ is a set of strings over a given alphabet $\mathcal{A}$. A *grammar* G for the language $\mathcal{L}$ consists of:

- A set T of *terminals* which are the letters of the alphabet $\mathcal{A}$ possibly augmented by the null-character ε.
- A set N of *nonterminals* which represent the syntactic categories of $\mathcal{L}$
- A set P of *production* or *derivation rules* which are used to derive the strings in $\mathcal{L}$. Each production consists of a nonterminal "head" that is produced and a string of zero or more nonterminal and terminals (the "body" of the production)
- A single nonterminal $S \in N$ that is designated as the start symbol.

The "dot-parenthesis" grammar for RNA, in the simplest case, can be written as $\mathrm{G}_0 = (T, N, P, s)$ with $T = \{(,), ., \varnothing\}$, $N = \{S\}$, $s = S$ and:

$$P = \left\{S \to S., S \to (S)S, S \to \varnothing\right\}, \tag{14}$$

where $\varnothing$ denotes the empty string. The grammar above is *context-free* since all productions are of the form $V \longrightarrow w$, where V is a nonterminal and w is a string consisting of terminals and/or nonterminals. The grammar generates strings of dots and balanced parenthesis; the parse trees of this grammar correspond to the secondary structures. More elaborate grammars can be designed that explicitly encode different types of loops or other substructures. In particular, the decompositions of the structure sets in Section 3.3 can be recast in terms of a grammar:

$$\begin{aligned} F &\longrightarrow uF|CF|\varnothing \\ C &\longrightarrow pL'\bar{p} \mid pLCL\bar{p} \mid pMN\bar{p} \\ M &\longrightarrow LC \mid MC \mid Mu \\ N &\longrightarrow Nu \mid C \\ L' &\longrightarrow uuuL \\ L &\longrightarrow uL \mid \varnothing. \end{aligned} \tag{15}$$

This grammar generates RNA sequences, while again the parse trees correspond to secondary structures (Figure 12). The terminal u denotes an unpaired base, while p and $\bar{p}$ is a shorthand for one of the six pairing combinations of bases. The start symbol F represents any structure, L stands for an unpaired sequence within a loop, M and N represent the left and right half of a multiloop. The production for L' enforces the minimum length of a hairpin loop.

Chomsky normal forms have only productions of the form $V \longrightarrow XY$ and $V \longrightarrow a$ with $V, X, Y \in N$ and $a \in T$. One can show that every context-free grammar can be converted to normal form, i.e. there is a CFG in normal form that produces the same language $\mathcal{L}$.

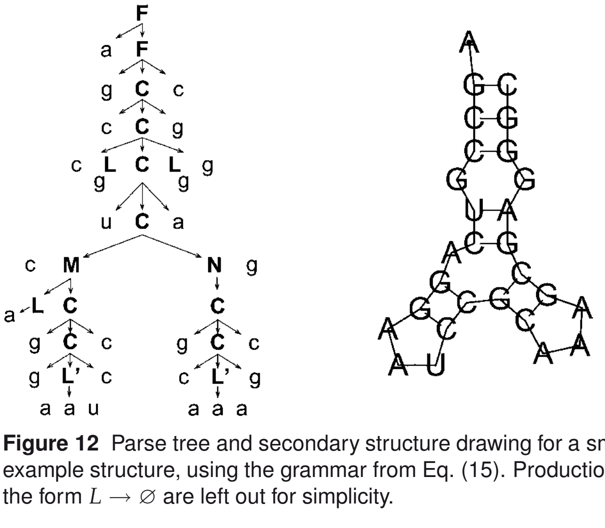


Figure 12 Parse tree and secondary structure drawing for a small example structure, using the grammar from Eq. (15). Productions of the form $L \longrightarrow \varnothing$ are left out for simplicity.

Given a CFG $G = (T, N, P, s)$ we obtain a *stochastic CFG* (*SCFG*) by assigning probabilities $\mathbb{P}(\alpha)$ to all productions $\alpha \in P$ such that $\sum_{\alpha \in P} \mathbb{P}(\alpha) = 1$ is satisfied.

The probabilities associated with the individual productions take on the role of the energy parameters in the previous sections. While the energy parameters must be measured directly, the values of $\mathbb{P}(\alpha)$ can be inferred from training sets of known sequence/structure pairs in a generic machine learning setting. Thus, they can, at least in principle, readily combine different sources of information that can be expressed probabilistically, such as an evolutionary model (derived from a comparative analysis of RNA sequences) and a biophysically motivated model of structure plausibility. In the following three subsection we briefly outline the basic techniques: finding the most likely parse-tree, computing the probability of a given word and the estimation of production probabilities from a given dataset. None of these algorithms is RNA specific; rather, they apply to any SCFG in Chomsky normal form.

5.2 Cocke–Younger–Kasami (CYK) Algorithm

The analog of the minimum free energy folding problem in the SCFG setting can be phrased in the following way: given a string $x \in \mathcal{L}$, find the most likely parse tree for x in a grammar G.

Under the assumption that G is in Chomsky normal form, there is an efficient (polynomial-time) solution to this question, the CYK algorithm [146].

Let $w(i, j, V)$ denote the likelihood of the most likely parse tree on the substring $x[i, \ldots, j]$ rooted at the nonterminal V. Clearly, we have $w(i, i, V) = \log \mathbb{P}(V \to x_i)$ for all i and V. For all larger substrings, $j > i$, we try all productions of the form $V \to XY$ and select the one that maximizes the likelihood. This immediately leads to the recursion

$$w(i, j, V) = \max_X \max_Y \max_{i \le k < j} \left[\log \mathbb{P}(V \to XY) + w(i, k, X) + w(k+1, j, Y) \right] \quad (16)$$

with the initialization $w(i, i, V) = \log \mathbb{P}(V \to x_i)$. The same type of backtracing approach as in the Nussinov algorithm can be used to explicitly recover the parse tree, which corresponds to the secondary structure of the RNA molecule.

5.3 Inside and Outside Algorithms

Instead of retrieving the most likely parse tree one may instead be interested in the probabilities of generating substrings in a particular way. In particular, let $p(i, j, V)$ be the probability that the "inside" substring $x[i, \ldots, j]$ is generated by the nonterminal V. Furthermore, let $q(i, j, V)$ be the probability that the "outside" substrings $x[1, \ldots, i-1] \cup x[j+1, \ldots, n]$ are generated from the start symbol S under the condition that (the parse subtree of) the subsequence $x[i,\ldots,j]$ is rooted at V. Conceptually, these quantities correspond to the partition functions inside and outside of a subsequence $x[i,\ldots,j]$. It is straightforward to derive the corresponding *inside recursion*:

$$p(i,j,V) = \sum_X \sum_Y \sum_{k=i}^{j} \mathbb{P}(V \to XY) p(i,k,X) p(k+1,j,Y), \tag{17}$$

which is initialized with $p(i,i,V) = \mathbb{P}(V \to x_i)$. The *outside recursion* consists of two parts, depending on whether the root V of the interior parse tree is the right or the left nonterminal in the previous production. This yields:

$$\begin{aligned} q(i,j,V) =& \sum_X \sum_Y \sum_{k<i} \mathbb{P}(Y \to XV) p(k,i-1,X) q(k,j,Y) \\ &+ \sum_X \sum_Y \sum_{k>j} \mathbb{P}(Y \to VX) q(i,k,Y) p(j+1,k,X), \end{aligned} \tag{18}$$

with the initial conditions $q(1,n,S) = 1$ and $q(1,n,X) = 0$ for all $X \in N \setminus \{S\}$. The probability to produce the sequence x is:

$$\mathbb{P}(x) = p(1,n,S) = \sum_X q(i,i,X) \mathbb{P}(X \to x_i). \tag{19}$$

5.4 Parameter Estimation

One problem with SCFG approaches is that the production probabilities have to be estimated from data. To this end, we compute the expected number $c(V)$ that V is used to parse x and the expected numbers $c(\alpha)$ that production α is used in the derivation of x. It is straightforward to derive:

$$\begin{aligned} c(V) &= \frac{1}{\mathbb{P}(x)} \sum_{i,j=1}^{n} p(i,j,V) q(i,j,V) \\ c(V \to a) &= \frac{1}{\mathbb{P}(x)} \sum_{i:x_i=a} q(i,i,V) \mathbb{P}(V \to a) \\ c(V \to XY) &= \frac{1}{\mathbb{P}(x)} \sum_{i,j=1}^{n} \sum_{k=i}^{j} q(i,j,V) p(i,k,X) p(k+1,j,Y) \mathbb{P}(V \to XY). \end{aligned} \tag{20}$$

Updated estimates for the production probabilities can thus be obtained as $\mathbb{P}'(\alpha) = c(\alpha)/c(V)$ for all $\alpha \in P$. The procedure is then repeated until $\sum_\alpha |\mathbb{P}'(\alpha) - \mathbb{P}(\alpha)| < \varepsilon$, where ε is a user-defined accuracy.

5.5 Algebraic Dynamic Programming

Algebraic Dynamic Programming (ADP) [36] was introduced to facilitate and systematize the development of dynamic programming algorithms. Conceptually, a dynamic programming algorithm consists of three components: a search space of candidate solutions (in our case RNA secondary structures), a scoring scheme (free energies, partition functions, etc.) and an objective function [minimize (energy), sum up (Boltzmann factors)]. The idea behind ADP is to separate these three aspects. For a comprehensive discussion of ADP in the context of bioinformatics we refer to Ref. [36]. We can give here only a very brief, qualitative sketch of the topic.

The search space is defined by a *yield grammar*, i.e. a tree grammar that generates a string language by mapping its terminal symbols at the leaves of the tree into sequences of symbols. A tree grammar is similar to a CFG, with terminal and nonterminal symbols, and productions where the right-hand sides are trees (formulas) from some underlying term algebra. Intuitively, first the search space is "constructed" by enumerating all candidate solutions. This is a parsing problem for which standard solutions, so-called tabulating yield parsers, exist. Scoring and choice are described in terms of an *evaluation algebra*, which is independent of the details of the search space.

The main advantage is that complex variants of folding problems can be implemented very easily. It suffices to modify the grammar to restrict the dynamic programming recursions to all canonical secondary structures, i.e. those that have no isolated base pairs. Conversely, the evaluation algebra can be changed easily. Once the energy model is implemented, one can change the choice function from minimizing energies to adding up Boltzmann factors or listing all structures within an energy range.

The restrictions of the search space can be quite dramatic. One can, for example, restrict oneself to saturated secondary structures, which consist solely of maximally extended stacking regions, i.e. no adjacent single-stranded nucleotides exist that could form a base pair and stack on top of a helix [26]. A particularly interesting application of the ADP framework is RNAshapes [37] which can be used to systematically generate (sub)optimal RNA structures belonging to distinct course-grained structural classes. For example, one can search for the most stable clover-leaf shaped secondary structure that can be formed by the input sequence.

5.6 Notes

Due to space restrictions we only gave a brief sketch of the SCFG approach to RNA secondary structures. A variety of implementations of SCFG-based algorithms are available for different purposes: pfold [74, 75] as an SCFG-approach to "folding an alignment" similar in spirit to the thermodynamics-based RNAalifold.

A general approach to computing suboptimal parse trees, similar in spirit to the backtracing of RNA secondary structures with suboptimal energies, is described in Ref. [70]. A systematic comparison of several alternative grammar models for RNA secondary structures showed that the actual performance of SCFGs can depend considerably on the details of the grammar being used [23].

A practical problem for the application of SCFGs is that one needs a grammar that is both unambiguous and in Chomsky normal form. The decomposition of Figure 6, for example, does not satisfy this requirement, because the last case in the second line, for example, requires nonterminals for the closing base pair as well as for the two enclosed multiloop components. Without discussing the details here, this creates problems in particular with the multiloop decomposition.

Sean Eddy's Infernal [25] creates a covariance model from local alignments and can be used to search a sequence database for sequences that are likely to be produced from this SCFG. Rsearch [73] aligns an RNA query to target sequences, using SCFG algorithms to score both secondary structure and primary sequence alignment simultaneously.

So-called pair SCFGs can be used to solve the combined folding and alignment problem in analogy to Sankoff's algorithm described in the next section, (e.g. Ref. [63]). The QRNA program [111] uses a pair SCFG to compute the probability that the substitution pattern in a pairwise alignment is derived from RNA secondary structure conservation. It has been used successfully to predict ncRNA candidates in *Escherichia coli* and *Saccharomyces cerevisiae* [94, 112]. Most recently, Pedersen and coworkers [102, 103] devised an SCFG-based algorithm for detecting conserved secondary structure motifs specifically within coding sequences. An SCFG-like approach to pseudoknotted structures can be found in Ref. [11].

6 Comparison of Secondary Structures

Many classes of functional RNA molecules, including tRNAs, rRNAs and many other "classical" ncRNAs, are characterized by highly conserved secondary structures, but little detectable sequence similarity. Reliable multiple alignments can therefore be constructed only when the shared structural features are taken into account. Since multiple alignments are used as input for many subsequent methods of data analysis, structure-based alignments are an indispensable necessity in RNA bioinformatics. This problem is far from being solved in a satisfactory way, both because the available approaches are computationally expensive and because little is known about the evolution of RNA at the structural level, and hence on the appropriate edit cost parameters.

6.1 String-based Alignments

The problem of comparing two structures Ψ_1 and Ψ_2 of the *same* RNA molecule is trivial. Since a secondary structures is simply a set of base pairs one may use, for example, the size of the symmetric difference between the two sets $|\Psi_1 \triangle \Psi_2|$ as a distance measure that is obviously a metric. In other words, we simply count the number of base pairs that occur in one of the structures, but not in both,

The question immediately becomes nontrivial, however, if we do not assume that the two structures have the same underlying sequence length, i.e. if we do not know *a priori* which sequence positions in the two molecules correspond to each other.

As we have seen, RNA secondary structures can be faithfully represented as strings over the alphabet $\{(,),.\}$. Clearly, we can use this string representation to compute a metric on secondary structures by means of standard sequence alignment methods, e.g. using the Needleman–Wunsch algorithm [99].

This approach can be generalized to a comparison of base pair probability matrices [7]. From the pairing probabilities of base i we construct a vector containing the probabilities of being paired upstream $p^{<}(i) = \sum_{j>i} P_{ij}$, downstream $p^{>}(i) = \sum_{j<i} P_{ji}$ or unpaired $p^{\circ}(i) = 1 - p^{<}(i) - p^{>}(i)$. The resulting profiles can be aligned by means of a standard string/profile alignment algorithm in $\mathcal{O}(n^2)$ time using:

$$\rho = \sqrt{p_A^{>} p_B^{>}} + \sqrt{p_A^{<} p_B^{<}} + \sqrt{p_A^{\circ} p_B^{\circ}}, \tag{21}$$

as the match score (or $1 - \rho$ as an edit cost). While this approach of "string-like alignments" is fast, it often produces misaligned pairs (Figure 13).

```
  Sequence alignment          Structure alignment
CAGUCUCAGGUGGUUGGGCU-       CAGUCUCAGGUGGUUG-GGCU
.((((.(((....)))))))-       .((((.(((....)))-))))
UAG-CUGAGGUG-UCGUGCUA       -UAGC-UGAGGUGUCGUGCUA
(((-((((....-))).))))       -((((-(((....))).))))
```

Figure 13 Sequence versus structure alignment. Compared to the structural alignment (right), the sequence alignment (from ClustalW) misaligns five of the seven base pairs.

6.2 Tree Editing

The string-based alignments above essentially use only the information whether a nucleotide is paired or unpaired, but neglect the connectivity information who pairs with whom. This limitation can be overcome by methods based on the tree representation of secondary structures. Of particular interest are tree editing and the related tree alignment, since they are still fast enough to be applicable to genome wide surveys. We present these approaches in detail here since there does not appear be a good textbook exposition of this topic.

The three most natural operations ("moves") that can be used to convert ordered trees (and, more generally, ordered forests) into each other are depicted in Figure 14:

(i) *Substitution* $(x \to y)$ consists of replacing a single vertex label x by another vertex label y.

(ii) *Insertion* $(\varnothing \to z)$ consists of adding a vertex z as a child of x, thereby making z the parent of a consecutive subsequence of children of x. A node z can also be inserted at the "top level", thereby becoming the root of a tree.

(iii) *Deletion* $(z \to \varnothing)$ consists of removing a vertex z, its children thereby become children of the parent x of z. Removing the root of a tree produces a forest in which the children of z become roots of trees.

Naturally, we associate a *cost* with each edit operation, which we will denote by $\gamma(x \to y)$, $\gamma(\varnothing \to z)$ and $\gamma(z \to \varnothing)$ for substitutions, insertions and deletions, respectively. We assume that γ is a metric on the extended alphabet $\mathcal{A} \cup \{\varnothing\}$. By using an appropriate alphabet of vertex labels, one can easily include sequence information in the cost function.

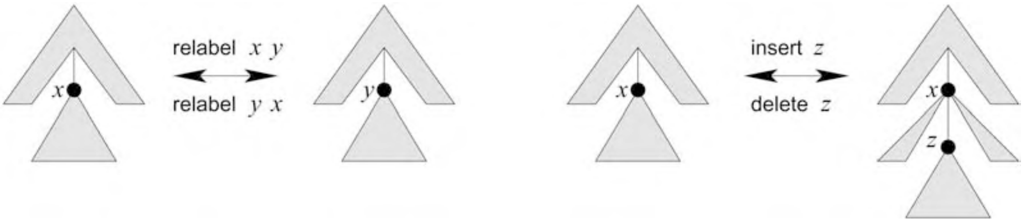


Figure 14 Elementary operations in tree editing

A sequence of moves that transforms a forest F_1 into a forest F_2 is known as an *edit script*. Its cost is the sum of the costs of edit operations in the script.

A *mapping* from F_1 to F_2 is a binary relation $M \in V(F_1) \times V(F_2)$ between the vertex sets of the two forests such that for pairs $(x,y), (x',y') \in M$ holds

(i) $x = x'$ if and only if $y = y'$ (one-to-one condition).

(ii) x is an ancestor of x' if and only if y is an ancestor of y' (ancestor condition).

(iii) x is to the left of x' if and only if y is to the left of y' (sibling condition).

By definition, for each $x \in F_1$ there is a unique "partner" in $y \in F_2$ such that $(x, y) \in M$ or there is no partner at all. In the latter case we write $x \in M_1'$. Analogously, we write $y \in M_2'$ if $y \in F_2$ does not have a partner in F_1. With each mapping we can associate the cost:

$$\gamma(M) = \sum_{(x,y)\in M} \gamma(x \to y) + \sum_{y \in M_2'} \gamma(\varnothing \to y) + \sum_{x \in M_1'} \gamma(x \to \varnothing). \tag{22}$$

Clearly, each edit operation gives rise to a corresponding mapping between the initial and the final tree. In the case of a substitution, all vertices have partners; in the case of insertion and deletion, there is exactly one vertex without partner.

Mappings are relations and hence they can be composed in a natural way. Consider three forests F_1, F_2 and F_3 and mappings M_1 from F_1 to F_2 and M_2 from F_2 to F_3. Then:

$$M_1 \circ M_2 = \left\{(x, z) \,\middle|\, \exists y \in V(F_2) \text{ such that } (x, y) \in M_1 \text{ and } (y, z) \in M_2\right\}, \tag{23}$$

is a mapping from F_1 to F_3. It is easy convince oneself that the cost function defined in Eq. (22) is subadditive under composition, $\gamma(M_1 \circ M_2) \leq \gamma(M_1) + \gamma(M_2)$. Using this result and the fact that every mapping can be obtained as a composition of edit operations one can show that the minimum cost mapping is equivalent to the minimum cost edit script [128].

For a given forest F we note by $F - x$ the forest obtained by deleting x and $F \setminus T(x)$ is the forest obtained from F by deleting with x all descendants of x. Note that $T(x) - x$ is the forest consisting of all trees whose roots are the children of x.

Now consider two forests F_1 and F_2, and let v_i be the root of the right-most tree in F_i, $i = 1, 2$ and an optimal mapping M. Apart from the trivial cases, in which one of the two forests is empty, we have to distinguish three cases. (i) v_2 has no partner in the optimal mapping. In this case, v_2 is inserted and the optimal mapping consists of an optimal mapping from F_1 to $F_2 - v_2$ composed with the insertion of v_2. (ii) v_1 has no partner. This corresponds to the deletion of v_1. (iii) both v_1 and v_2 have partners. In this case $(v_1, v_2) \in M$.

To see this, one can argue as follows. Suppose $(v_1, h) \in M$, $h \neq v_2$ and $(k, v_2) \in M$. By the one-to-one condition, $k \neq v_1$. By the sibling condition, if v_1 is to the right of k, then h must be to the right of v_2. If v_1 is a proper ancestor of k, then h must be a proper ancestor of v_2 by the ancestor condition. Both cases are impossible, however, since both v_1 and v_2 are by construction right-most roots.

For each of the three cases it is now straightforward to recursively compute the optimal cost of M. We arrive directly at the dynamic programming recursion:

$$D(F_1, F_2) = \min \begin{cases} D(F_1 - v_1, F_2) + \gamma(v_1 \to \varnothing), \\ D(F_1, F_2 - v_2) + \gamma(\varnothing \to v_2), \\ D(T(v_1) - v_1, T(v_2) - v_2) + \\ \qquad\qquad D(F_1 \setminus T(v_1), F_2 \setminus T(v_2)) + \gamma(v_1 \to v_2). \end{cases} \tag{24}$$

which allows us to compute the tree edit distance $D(F_1, F_2)$ from smaller subproblems. The initialization is the distance between $D(\emptyset, \emptyset) = 0$ of two empty forests. In the cases where one of the two forests is empty, Eq. (24) reduces to $D(\emptyset, F_2) = D(\emptyset, F_2 - v_2) + \gamma(\varnothing \to v_2)$ and $D(F_1, \emptyset) = D(F_1 - v_1, \emptyset) + \gamma(v_1 \to \varnothing)$.

One can show that the time complexity of this algorithm is bounded by $\mathcal{O}(|F_1|^2|F_2|^2)$. Various more efficient implementations exist (see in particular Refs. [72, 148]). A detailed performance analysis of the algorithm by Zhang and Shasha [148] is given in Ref. [24].

A common feature of all tree representations discussed above is that each subtree $T(x)$ rooted at a vertex x corresponds to an interval I_x of the underlying RNA sequence. We can thus regard every pair (v_1, v_2) as a prescription to match up the intervals I_{v_1} with J_{v_2} between the two input sequences. In particular, if v_1 and v_2 are leaves in the forests F_1 and F_2, then they correspond to individual bases. Interior nodes serve as delimiters of intervals in Giegerich's encoding, while they correspond to base pairs in the encoding used in the Vienna RNA Package. In either case, one can derive all (mis)matches directly from M. The sibling and ancestor properties of M guarantee that (mis)matches preserve the order in which they appear on the RNA sequence. All other nucleotides, i.e. those that correspond to vertices $v_1 \in M_1'$ and $v_2 \in M_2'$, are deleted or inserted, respectively, in the appropriate positions. Every mapping M therefore implies a (canonical) pairwise alignment $\mathbb{A}(M)$ of the underlying sequences.

6.3 Tree Alignments

An alternative way of defining the difference of two forests is using *tree alignments* [69]. Consider a forest G with vertex labels taken from $(\mathcal{A} \cup \{-\}) \times (\mathcal{A} \cup \{-\})$. Then we obtain restrictions $\pi_1(G)$ and $\pi_2(G)$ by considering only the first or the second coordinate of the labels, respectively, and by then deleting all nodes that are labeled with the gap character "–" (Figure 15). We say that G is an alignment of the two forests F_1 and F_2 if $F_1 = \pi_1(G)$ and $F_2 = \pi_2(G)$. Naturally, we score the alignment G by adding up the costs of the

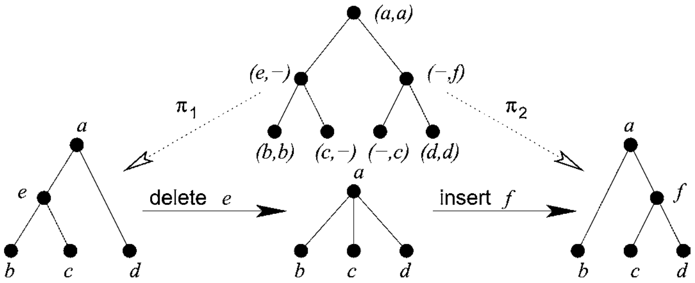


Figure 15 Alignment of two forests F_1 and F_2 and a mapping from F_1 to F_2 that cannot be derived from an alignment.

label pairs:

$$\gamma(G) = \sum_{(v_1,v_2)\in G} \gamma(v_1 \to v_2), \tag{25}$$

where a pair $(v_1, -)$ corresponds to the edit operation $(v_1 \to \varnothing)$. On the other hand, an alignment G defines a mapping M_G from F_1 to F_2 by setting $(v_1, v_2) \in M_G$ iff $(v_1, v_2) \in G$ and neither v_1 nor v_2 is a gap character. One easily verifies that the three defining properties of mapping are satisfied. Furthermore, it follows that $\gamma(M_G) = \gamma(G)$ as pair of the form $(v_1, -)$ and $(-, v_2)$ corresponds to deletion and insertion operations, respectively. Note that, in the special case of two a totally disconnected forests, the problem reduces to ordinary sequence alignment with additive gap costs.

However, as the example in Figure 15 shows, not all mappings derive from alignments. It follows, therefore, that the minimum cost alignment is more costly than the minimum cost edit script, in general.

In order to compute the optimal alignment, let us first investigate the decomposition of an alignment at a particular (mis)match (v_1, v_2) or in/del $(-, v_2)$ or $(v_1, -)$. We will need a bit of notation (Figure 16). Let F be an ordered forest. By $i : F$ we denote the subforest consisting of the first i trees, while $F : j$ denotes the subforest starting with the $j + 1$-th tree. By $F^{\downarrow}$ we denote forest consisting of the children trees of the root $v = r_F$ of the first tree in F. $F^{\rightarrow} = F : 1$ is the forest of the right siblings trees of F.

Now consider an alignment A of two forests F_1 and F_2. Let $a = r_A$ be the root of its first tree. We have either:

(i) $a = (v_1, v_2)$. Then $v_1 = r_{F_1}$ and $v_r = r_{F_2}$; $A^{\downarrow}$ is an alignment of $F_1^{\downarrow}$ and $F_2^{\downarrow}$; $A^{\rightarrow}$ is an alignment of $F_1^{\rightarrow}$ and $F_2^{\rightarrow}$.

(ii) $a = (-, v_2)$. Then $v_2 = r_{F_2}$; for some k, $A^{\downarrow}$ is an alignment of $k : F_1$ and $F_2^{\downarrow}$ and $A^{\rightarrow}$ is an alignment of $F_1 : k$ with $F_2^{\rightarrow}$.

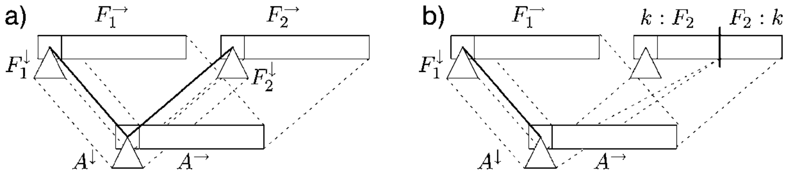


Figure 16 Decomposition of tree alignments. (a) In the match case the subtrees $F_1^{\downarrow}$ and $F_2^{\downarrow}$ are aligned to form $A^{\downarrow}$ and, correspondingly, the sibling subforests $F_1^{\rightarrow}$ and $F_2^{\rightarrow}$ must be aligned to yield $A^{\rightarrow}$. (b) In the deletion case the subforest $F_1^{\downarrow} - v_1$ must be aligned with a part $k : F_2$ of the second forest. $F_1^{\rightarrow}$ then must be aligned with remainder of $F_2 : k$ of the top-level trees of F_2. The insertion case is analogous to the deletion case, with the roles of F_1 and F_2 exchanged.

(iii) $a = (v_1, -)$. Then $v_1 = r_{F_1}$; for some k, $A^{\downarrow}$ is an alignment of $F_1^{\downarrow}$ and $k : F_2$ and $A^{\rightarrow}$ is an alignment of $F_1^{\rightarrow}$ with $F_2 : k$.

See Figure 16 for a graphical representation.

Let $S(F_1, F_2)$ be the optimal score of an alignment of the forests F_1 and F_2. For easier comparison with the tree-editing algorithm in the previous section we formulate the problem here as a minimization problem. One can, however, just as well maximize appropriate similarity scores. The three cases discussed above and in Figure 16 imply the following dynamic programming recursion:

$$S(F_1, F_2) = \min \begin{cases} S(F_1^{\downarrow}, F_2^{\downarrow}) + S(F_1^{\rightarrow}, F_2^{\rightarrow}) + \gamma(v_1 \to v_2) \\ \min_k S(k : F_1, F_2^{\downarrow}) + S(F_1 : k, F_2^{\rightarrow}) + \gamma(\varnothing \to r_{F_2}) \\ \min_k S(F_1^{\downarrow}, k : F_2) + S(F_1^{\rightarrow}, F_2 : k) + \gamma(r_{F_1} \to \varnothing). \end{cases} \tag{26}$$

In the special cases where one of the forests is empty this reduces to $S(\emptyset, F_2) = S(\emptyset, F_2^{\downarrow}) + S(\emptyset, F_2^{\rightarrow}) + \gamma(\varnothing \to r_{F_2})$ for the insertion case, and $S(F_1, \emptyset) = S(F_1^{\downarrow}, \emptyset) + S(F_1^{\rightarrow}, \emptyset) + \gamma(r_{F_1} \to \varnothing)$ for the deletion case. The initial condition is again $S(\emptyset, \emptyset) = 0$.

In order to estimate the resource requirements for this algorithm, we observe that we have to consider only those subforests of F_1 and F_2 that consist of trees rooted at an uninterrupted interval of sibling nodes. These forests have been termed *closed subforests* in Ref. [51]. If d_i is the maximum of the number of trees and the numbers of children of the nodes in F_i, we see that there are at most $\mathcal{O}(d_i^2)$ closed subforests at each node and hence at most $\mathcal{O}(|F_1|\,|F_2| d_1^2 d_2^2)$ entries $S(F_1, F_2)$ need to be computed, each of which requires $\mathcal{O}(d_1 + d_2)$ operations, i.e. tree alignments can be computed in polynomial time. A compact, memory-efficient encoding of the subforests is described in detail in Ref. [51], where a careful analysis shows that pairwise tree alignments can be computed in $\mathcal{O}(|F_1| d_1\, |F_2| d_2)$ space and $\mathcal{O}(|F_1|\,|F_2|\, d_1\, d_2 (d_1 + d_2))$ time.

6.4 The Sankoff Algorithm and Variants

David Sankoff described an algorithm that simultaneously allows the solution of the structure prediction and the sequence alignment problem [114]. The basic idea is to search for a maximal secondary structure that is common to two RNA sequences. Given a score $\sigma_{ij,kl}$ for the alignment of the base pairs (i, j) and (k, l) from the two sequences (as well as gap penalties γ and scores α_{ik} for matches of unpaired positions) we compute the optimal alignment recursively from alignments of the subsequences $x[i, \ldots, j]$ and $y[k, \ldots, l]$. Let $S_{ij,kl}$ be the score of the optimal alignment of these fragments. We have:

$$\begin{aligned} S_{ij;kl} = \max \Big\{ & S_{i+1,j;kl} + \gamma, S_{ij;k+1,l} + \gamma, S_{i+1,j;k+1,l} + \alpha_{ik}, \\ & \max_{(p,q) \text{ paired}} \Big\{ S_{i+1,p-1;k+1,q-1} + \sigma_{ij,pq} + S_{p+1,j;q+1,l} \Big\} \Big\} . \end{aligned} \tag{27}$$

Backtracing is just as easy as in the RNA folding case. Only now π is a partial alignment of two structures and we insert aligned positions instead of positions in individual structures. More precisely we have to insert individual columns or pairs of columns of the form:

$$\pi \blacktriangleleft \begin{pmatrix} i. \\ - \end{pmatrix} \quad \pi \blacktriangleleft \begin{pmatrix} - \\ j. \end{pmatrix} \quad \pi \blacktriangleleft \begin{pmatrix} i. \\ j. \end{pmatrix} \qquad \pi \blacktriangleleft \begin{pmatrix} i(& & j) \\ p(& , & q) \end{pmatrix} , \tag{28}$$

into a growing partial alignment π, just as we insert unpaired bases or base pairs in the backtracing of the folding algorithm in Section 3.2.

This algorithm is computationally very expensive, however. It requires $\mathcal{O}(n^4)$ memory and $\mathcal{O}(n^6)$ CPU time. Currently available software packages such as foldalign [39,65] and dynalign [92] therefore implement only restricted versions. The simple, maximum matching style version is used in pmcomp [52] as an approach to comparing base pairing probability matrices.

6.5 Multiple Alignments

Pairwise alignment methods, be they for sequences or structures, can be readily generalized to alignments of many objects. Usually, it is too costly to compute optimal multiple alignments exactly and one therefore resorts to heuristics such as progressive multiple alignments. pmmulti [52], for example, produces multiple structural alignments in the context of the Sankoff algorithm by calling pmcomp for pairwise alignments. For tree alignments, the RNAforester programs can be used to compute both pairwise and progressive multiple alignments.

As we have seen above, the mappings produced by tree editing do not correspond to tree alignments in general. These methods can therefore not be used for comparing multiple structures. The edit scripts can, however, be interpreted in terms of a sequence alignment. One may therefore still use these methods as the starting point for multiple sequence alignments. This is the central idea of the MARNA program [121] which uses pairwise structural alignments as input to the multiple alignment program T-Coffee [100].

6.6 Notes

Various variants, specializations and generalizations of the tree-editing approach have been described in recent years. Examples include efficient algorithms for similar trees [67] and with simplified edit cost models [120]. Tree-editing with restricted mappings M satisfying stronger requirements on structural conservation are described in Ref. [147]. Let $\mathrm{lca}(a,b)$ denote the "last common ancestor" of a and b. For all $(x',x''),(y',y''),(z',z'') \in M$ holds: $\mathrm{lca}(x',y')$ is a proper ancestor of z' if and only if $\mathrm{lca}(x'',y'')$ is a proper ancestor of z''. Other variants of tree edit distances have also been discussed (e.g. Ref. [132]). A tree-edit model for RNA that allows additional "node-fusion" and "edge-fusion" events is described in Ref. [4].

More general edit models with application to RNA structures are described in Refs. [68, 88], an alignment distance for pseudoknotted structures can be found in Ref. [9].

A very different approach to the pairwise comparison of RNA structures, with or without pseudoknots, converts the RNA alignment problem into an integer programming problem [79]. Recently, efficient algorithms based on Lagrangian relaxation have been developed [6], that have helped to make the performance of this approach comparable to other methods.

A partition function version of the Sankoff algorithm, which can be used to compute the probabilities of all possible (mis)matches in a structural alignment of two RNA base pairing probability matrices is described in Ref. [53]. RNA structure comparison can also be recast in the SCFG framework [64]. The corresponding pair-SCFG algorithms correspond to the Sankoff algorithm.

7 Kinetic Folding

7.1 Folding Energy Landscapes

The folding dynamics of a particular RNA molecule can also be studied successfully within the framework of secondary structures. The folding process is determined by the energy landscape [or potential energy surface (PES) in the terminology of theoretical chemistry]. Instead of considering all possible spatial conformations, it is meaningful to partition the conformation space into sets of conformations that belong to a given secondary structure. Instead of a smooth surface defined on a space of real-valued coordinate vectors we are therefore dealing with a landscape on a complex graph [107]. The vertices of this graph are the secondary structures that can be formed by the given RNA sequence, the edges are determined by a rule specifying which structures can be interconverted and the height of the landscape at a structure x is its free energy $E(x)$. Typically, one considers a "move set" that allows the insertion and deletion of single base pairs. In addition, a shift-move that changes (i,j) to (i,k) or (h,j) is sometimes included [30]. Further coarse-grainings of this landscape can be achieved, e.g. by considering secondary structures as composed of stacks instead of individual base pairs.

7.2 Kinetic Folding Algorithms

Several groups have designed kinetic folding algorithms for RNA secondary structures, mostly in an attempt to obtain more accurate predictions or in order to include pseudoknots (see e.g. Refs. [2, 41, 89, 95, 126]). Only a few papers have attempted to reconstruct folding pathways [42, 50, 124]. These algorithms generally operate on a list of all possible helices and consequently use move sets that destroy or form entire helices in a single move. Such a move set can introduce large structural changes in a single move and, furthermore, *ad hoc* assumptions have to be made about the rates of helix formation and disruption. A more local move set is, therefore, preferable if one hopes to observe realistic folding trajectories.

The process of kinetic folding itself can be modeled as homogeneous Markov chain. The probability p_x that a given RNA molecule will have the secondary structure x at time t is given by the master equation:

$$\frac{\mathrm{d}p_x}{\mathrm{d}t} = \sum_{y \in X} r_{xy} p_y(t)\,, \tag{29}$$

where r_{xy} is the rate constant for the transition from secondary structure y to secondary structure x in the deterministic description [38]. The transition state model dictates an expression of the form:

$$r_{yx} = r_0 \mathrm{e}^{-\frac{E^{\neq}_{yx} - E(x)}{RT}} \quad \text{for } x \neq y \quad \text{and} \quad r_{xx} = -\sum_{y \neq x} r_{yx}, \tag{30}$$

where the transition state energies $E^{\neq}_{yx}$ must be symmetric, $E^{\neq}_{yx} = E^{\neq}_{xy}$, and r_0 is a scaling constant. In the simplest case one can use:

$$E^{\neq}_{yx} = \max\{E(x), E(y)\}. \tag{31}$$

For short sequences or very restricted subsets of conformations Eq. (29) can be solved exactly or integrated numerically [126]. Solving the master equation for larger conformation spaces is out of the question. In such cases the dynamics can be obtained by simulating the Markov chain directly by a rejection-less Monte Carlo algorithm [27] and sampling a large number of trajectories.

7.3 Approximate Folding Trajectories and Barrier Trees

An alternative approach to the direct simulation of the master equation (29) starts with a more detailed analysis of folding energy landscape. Let us start with a few definitions.

A conformation x is a global minimum if $E(x) \leq E(y)$ for all $y \in X$ and a local minimum if $E(x) \leq E(y)$ for all neighbors y of x. The energy $\hat{E}$ of the lowest saddle point separating two local minima x and y is:

$$\hat{E}[x,y] = \min_{\mathbf{p} \in \mathbb{P}_{xy}} \max_{z \in \mathbf{p}} E(z), \tag{32}$$

where $\mathbb{P}_{xy}$ is the set of all paths $\mathbf{p}$ connecting x and y by a series of consecutive transformations taken from the move set. If the energy function is nondegenerate then there is a unique saddle point $s = s(x,y)$ connecting x and y characterized by $E(s) = \hat{E}[x,y]$. To each saddle point s there is a unique collection of conformations $B(s)$ that can be reached from s by a path along which the energy never exceeds $E(s)$. In other words, the conformations in $B(s)$ are mutually connected by paths that never go higher than $E(s)$. This property warrants to call $B(s)$ the *basin of attraction* below the saddle s.

Two situations can arise for any two saddle points s and s' with energies $E(s) < E(s')$. Either the basin of s is a "subbasin" of $B(s')$ or the two basins are disjoint. This property arranges the local minima and the saddle points in a unique hierarchical structure which is conveniently represented as a tree, termed a *barrier tree* (Figure 17).

An efficient flooding algorithm [31] can be used to identify local minima and saddle points starting, for example, from the complete list of suboptimal secondary structures produced by the RNAsubopt program [144]. Consider a stack Σ which initially contains all secondary structures in the order of ascending energy. We pop the element z from the top of Σ and check which of its neighbors we have already seen before, i.e. which of its neighbors have a lower energy. There are three cases:

(i) z has no neighbor with lower energy, then it is a local minimum, i.e. a new leaf of the barrier tree.

(ii) z has only lower energy neighbors that all belong the same basin, say $B(x)$. Then z itself also belongs to $B(x)$.

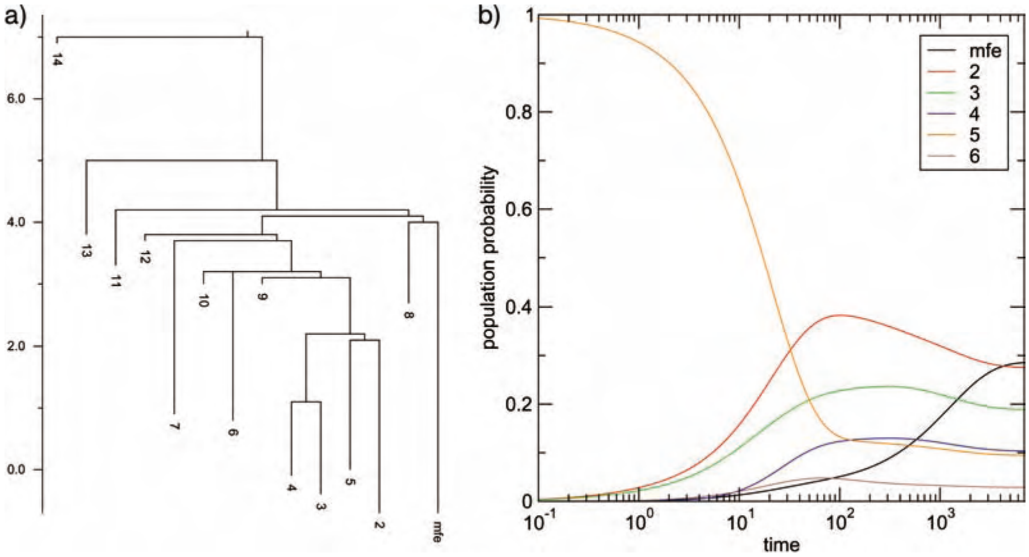


Figure 17 (a) Barrier tree of short artificial sequence UAUGCUGCGGCCUAGGC. The leaves of the tree are the local minima of the energy landscape. (b) Folding kinetics from the open structure. Population density p_α for the basin containing the local minimum α is shown for the six largest basins as a function of time.

(iii) z has lower-energy neighbors in two or more different basins. In this case z is the saddle point separating these basins, i.e. an interior vertex of the barrier tree. For the subsequent computation we now unify all basins connected by z into a new basin $B(z)$ and remove its subbasins from the list of "active" basins.

At the end, we are left with the barrier tree of the landscape. As a byproduct we also obtain the assignment of each secondary structure to its basin $B(x)$. Instead of searching through the list of all previously encountered structures it is more efficient to generate all neighbors of z and to check whether they have already been seen before by means of a hash-table lookup. The procedure thus runs in $\mathcal{O}(LD)$ time, where L is the length of the list of structures and D is the maximal number of moves that can be applied to a secondary structure. The barriers program implements this algorithm [31].

A description of the energy landscape or the dynamics of an RNA molecule based on all secondary structures is feasible only for very small sequences. We therefore need to coarse-grain the representation of the energy landscape. Let $\mathbf{\Pi} = \{\alpha, \beta, \dots\}$ be a partition of the state space. The classes of such a partition are *macrostates*. As a concrete example consider the partition of X defined by the gradient basins $\mathcal{B}(z)$ of the local energy minima. To each macrostate α we can assign the partition function:

$$Z_\alpha = \sum_{x\in\alpha} \mathrm{e}^{-E(x)/RT}, \tag{33}$$

and the corresponding free energy:

$$G(\alpha) = -RT \ln Z_\alpha. \tag{34}$$

The transition rates between macrostates can be obtained at least approximately from the elementary rate constants using the assumption that the random process is equilibrated within each macrostate [143]. Then:

$$r_{\beta\alpha} = \sum_{y\in\beta}\sum_{x\in\alpha} r_{yx} \frac{\mathrm{e}^{-E(x)/RT}}{Z_\alpha} \quad \text{for } \alpha \neq \beta. \tag{35}$$

We can use the transition state model to define the free energies of the transition state $G^{\neq}$ by setting:

$$r_{\beta\alpha} = r_0 \mathrm{e}^{-\frac{G^{\neq}_{\beta\alpha} - G(\alpha)}{RT}}. \tag{36}$$

A short computation then yields:

$$G^{\neq}_{\beta\alpha} = -RT \ln \sum_{y\in\beta}\sum_{x\in\alpha} \mathrm{e}^{-\frac{E^{\neq}_{yx}}{RT}}, \tag{37}$$

as one would expect.

In practice one can compute $r_{\beta\alpha}$ "on the fly" while executing the barriers program if two conditions are satisfied: (i) for each x we can efficiently determine to which macrostate it belongs and (ii) the double sum in Eq. (35) needs to be evaluated only for pairs of neighboring conformations (x, y). Condition (i) is easily satisfied for each of the gradient basins: in each step of the barriers algorithm all neighbors y of the newly added structures x that have a smaller energy have already been processed. Condition (ii) is satisfied by construction of the microscopic transition rates r_{xy}, which vanish unless x is a neighbor of y. In the case of short sequence, both the microscopic model and the macro-state model can be solved exactly. In many cases (e.g. Figure 17) the macro-state model provides a very good approximation of the dynamics.

7.4 RNA Switches

Some RNA molecules exhibit two meta-stable conformations, whose equilibrium can be shifted easily by external events, such as binding of another molecule. This can be used to regulate gene expression, when the two mutually exclusive alternatives correspond to an active and in-active conformation of the transcript. The best known example of such behavior are the riboswitches [133] found in the $5'$ untranslated regions of bacterial mRNAs, where the conformational change is triggered by binding of a small organic molecule.

Molecules that may be RNA switches can be recognized by inspection of the barrier tree, but this is feasible only for rather short sequences. The paRNAss program [134] instead uses a sample of suboptimal structures, and computes for every pair of structures "morphological" distance (e.g. tree edit distance) and a simple estimate of the energy barrier. The structures are then clustered according to these two measures, RNA switches are expected to exhibit two well separated clusters.

Interestingly, for any two secondary structures there exist sequences that are compatible with both structures, i.e. that can form both structures in principle [106]. If both structures are reasonably stable, it is not hard to design switching sequences with these two structures as stable conformations [28].

7.5 Notes

The analysis of landscapes becomes technically more complicated when structures, in particular adjacent structures, may have the same energy. In this case there is no unique definition of gradient basins and a variety of concepts, all related to saddle points, have to be distinguished (see Ref. [31] for further details).

The notion of barrier trees can be generalized to multivalued landscapes, which arise, for example, in the context of multiobjective optimization problems with conflicting constraints [122].

A computationally simpler alternative to the macrostate approach for transition rate is to assume an Arrhenius law $r_{\beta\alpha} \sim \exp(E^{\neq}/RT)$ and to approximate the transition state energy $E^{\neq}$ by the energy of the saddle point between the local minima α and β [143]

A generalization of the "intersection theorem" characterizes sets of more than two secondary structures that can be realized simultaneously by a common RNA sequence [28]. This observation can be used as a starting point for computational designs of switches with multiple states [1].

8 Concluding Remarks

Secondary structure drives the RNA-folding process, arguably even more so than it is the case for proteins. This renders the prediction of RNA secondary structure highly relevant for the prediction of RNA structure and analysis, in general. As this chapter shows, the field of RNA structure prediction is comparatively well developed. As a matter of fact, it is one of the fields in bioinformatics that benefits most comprehensively from algorithmic methods derived from computer science. The comparatively technical makeup of this chapter is a mirror of this phenomenon. Notably, very different questions, which in the protein world require different mathematical models, can be described and analyzed in the RNA case at the level of secondary structures: the thermodynamics of folding as well as the thermodynamics of RNA–RNA interactions are accessible via the same parameters and the same algorithms that can also be used to compute consensus structures in an evolutionary context or to investigate the dynamics of the folding process itself.

With the increased importance of RNA in biology, in general (consider, for instance, the recent surge in work on RNA interference (see also Chapter 45) and in the analysis of structural aspects of mRNA in the context of gene regulation), RNA secondary structure prediction is rapdily becoming an obligatory tool in the arsenal of bioinformatics analysis methods.

References

1 ABFALTER, I., C. FLAMM, AND P. F. STADLER. 2003. Design of multi-stable nucleic acid sequences. In Proc. Proc. German Conf. Bioinformatics, München, 1–7.

2 ABRAHAMS, J. P., M. VAN DEN BERG, E. VAN BATENBURG, AND C. PLEIJ. 1990. Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. Nucleic Acids Res. **18**: 3035–44.

3 AKUTSU, T.. 2001. Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. Discr. Appl. Math. **104**: 45–62.

4 ALLALI, J. AND M.-F. SAGOT. 2005. A new distance for high level RNA secondary structure comparison. IEEE/ACM Trans. Comp. Biol. Bioinf. **2**: 3–14.

5 ANDRONESCU, M., Z. ZHANG, AND A. CONDON. 2005. Secondary structure prediction of interacting RNA molecules. J. Mol. Biol. **345**: 987–1001.

6 BAUER, M. AND G. KLAU. 2004. Structural alignment of two RNA sequences with Lagrangian relaxation. Int. Symp. on Algorithms and Computation. Hong Kong: 113–25.

7 BONHOEFFER, S., J. S. MCCASKILL, P. F. STADLER, AND P. SCHUSTER. 1993. RNA multi-structure landscapes. a study based on temperature dependent partition functions. Eur. Biophys. J. **22**: 13–24.

8 BONNET, E., J. WUYTS, P. ROUZÉ, AND Y. VAN DE PEER. 2004. Evidence that microRNA precursors, unlike other non-coding RNAs, have lower folding free energies than random sequences. Bioinformatics **20**: 2911–17.

9 BRINKMEIER, M.. 2005. Structural alignments of pseudo-knotted RNA-molecules in polynomial time. Technical Report, TU Ilmenau.

10 BROWN, J. W., J. M. NOLAN, E. S. HAAS, M. A. T. RUBIO, F. MAJOR, AND N. R. PACE. 1996. Comparative analysis of ribonuclease P RNA using gene sequences from natural microbial populations reveals tertiary structural elements. Proc. Natl Acad. Sci. USA **93**: 3001–6.

11 CAI, L., R. L. MALMBERG, AND Y. WU. 2003. Stochastic modeling of RNA

pseudoknotted structures: a grammatical approach. Bioinformatics **19**(Suppl. 1): i66–73.

12 Cannone, J. J., S. Subramanian, M. N. Schnare, et al. 2002. The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. BMC Bioinformatics **3**: 2.

13 Chen, W. Y. C., E. Y. P. Deng, and R. R. X. Du. 2005. Reduction of *m*-regular noncrossing partitions. Eur. J. Comb. **26**: 237–43.

14 Chiu, D. K. and T. Kolodziejczak. 1991. Inferring consensus structure from nucleic acid sequences. CABIOS **7**: 347–52.

15 Clote, P.. 2005. An efficient algorithm to compute the landscape of locally optimal RNA secondary structures with respect to the Nussinov–Jacobson energy model. J. Comput. Biol. **12**: 83–101.

16 Condon, A., B. Davy, B. Rastegari, S. Zhao, and F. Tarrant. 2004. Classifying RNA pseudoknotted structures. Theor. Comput. Sci. **320**: 35–50.

17 Cristianini, N. and J. Shawe-Taylor. 2000. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK.

18 De Rijk, P. and R. De Wachter. 1997. RnaViz, a program for the visualisation of RNA secondary structure. Nucleic Acids Res. **25**: 4679–84.

19 Deutsch, E. and L. W. Shapiro. 2002. A bijection between ordered trees and 2-Motzkin paths and its many consequences. Discr. Math. **256**: 655–70.

20 Dimitrov, R. A. and M. Zuker. 2004. Prediction of hybridization and melting for double-stranded nucleic acids. Biophys. J. **87**: 215–26.

21 Ding, Y., C. Chan, and C. Lawrence. 2004. Sfold web server for statistical folding and rational design of nucleic acids. Nucleic Acids Res. **32**: W135–41.

22 Dirks, R. and N. Pierce. 2003. A parition function algorithm for nucleic acid secondary structure including pseudoknots. J. Comput. Chem. **24**: 1664–77.

23 Dowell, R. D. and S. R. Eddy. 2004. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. BMC Bioinformatics **5**: 71.

24 Dulucq, S. and L. Tichit. 2003. RNA secondary structure comparison: exact analysis of the Zhang–Shasha tree-edit algorithm. Theor. Comput. Sci. **306**: 471–84.

25 Eddy, S.. 2002. A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. BMC Bioinformatics **3**: 18.

26 Evers, D. J. and R. Giegerich. 2001. Reducing the conformation space in RNA structure prediction. Proc. German Conf. on Bioinformatics, Braunschweig. 118–24.

27 Flamm, C., W. Fontana, I. Hofacker, and P. Schuster. 2000. RNA folding kinetics at elementary step resolution. RNA **6**: 325–38.

28 Flamm, C., I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl. 2001. Design of multi-stable RNA molecules. RNA **7**: 254–65.

29 Flamm, C., I. L. Hofacker, and . P. F. Stadler. 2004. Computational chemistry with RNA secondary structures. Kemija u industriji **53**: 315–22.

30 Flamm, C., I. L. Hofacker, and P. F. Stadler. 1999. RNA *in silico*: the computational biology of RNA secondary structures. Adv. Complex Syst. **2**: 65–90.

31 Flamm, C., I. L. Hofacker, P. F. Stadler, and M. T. Wolfinger. 2002. Barrier trees of degenerate landscapes. Z. Phys. Chem. **216**: 155–73.

32 Fontana, W., D. A. M. Konings, P. F. Stadler, and P. Schuster. 1993. Statistics of RNA secondary structures. Biopolymers **33**: 1389–404.

33 Fontana, W., P. F. Stadler, E. G. Bornberg-Bauer, et al. 1993. RNA folding landscapes and combinatory landscapes. Phys. Rev. E **47**: 2083–99.

34 Gautheret, D., F. Major, and R. Cedergren. 1990. Pattern searching/alignment with RNA primary and secondary structures: an effective descriptor for tRNA. Comput. Appl. Biosci. **6**: 325–31.

35 Gibson, A., V. Gowri-Shankar, P. G. Higgs, and M. Rattray. 2005. A comprehensive analysis of mammalian mitochondrial genome base composition and improved phylogenetic methods. Mol. Biol. Evol. **22**(2): 251–64.

36 Giegerich, R.. 2000. A systematic approach to dynamic programming in bioinformatics. Bioinformatics **16**: 665–77.

37 Giegerich, R., B. Voss, and M. Rehmsmeier. 2004. Abstract shapes of RNA. Nucleic Acids Res. **32**: 4843–51.

38 Gillespie, D. T.. 1976. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. J. Comput. Phys. **22**: 403.

39 Gorodkin, J., L. J. Heyer, and G. D. Stormo. 1997. Finding the most significant common sequence and structure motifs in a set of RNA sequences. Nucleic Acids Res. **25**: 3724–32.

40 Gräf, S., D. Strothmann, S. Kurtz, and G. Steger. 2001. HyPaLib: a database of RNAs and RNA structural elements defined by hybrid patterns. Nucleic Acids. Res. **29**: 196–98.

41 Gultyaev, A. P.. 1991. The computer simulation of RNA folding involving pseudoknot formation. Nucleic Acids Res. **19**: 2489–93.

42 Gultyaev, A. P., van Batenburg, and C. W. A. Pleij. 1995. The computer simulation of RNA folding pathways using an genetic algorithm. J. Mol. Biol. **250**: 37–51.

43 Gultyaev, A. P., F. H. D. van Batenburg, and C. W. A. Pleij. 1999. An approximation of loop free energy values of RNA H-pseudoknots. RNA **5**: 609–17.

44 Gutell, R. R., A. Power, G. Z. Hertz, E. J. Putz, and G. D. Stormo. 1992. Identifying constraints on the higher-order structure of RNA: continued development and application of comparative sequence analysis methods. Nucleic Acids Res. **20**: 5785–95.

45 Gutell, R. R. and C. R. Woese. 1990. Higher order structural elements in ribosomal RNAs: Pseudo-knots and the use of noncanonical pairs. Proc. Natl Acad. Sci. USA **87**: 663–67.

46 Han, K. and Y. Byun. 2003. PSEUDOVIEWER2: Visualization of RNA pseudoknots of any type. Nucleic Acids Res. **31**: 3432–40.

47 Han, K., D. Kim, and H. J. Kim. 1999. A vector-based method for drawing RNA secondary structure. Bioinformatics **15**: 286–97.

48 Han, K. and H.-J. Kim. 1993. Prediction of common folding structures of homologous RNAs. Nucleic Acids Res. **21**: 1251–57.

49 Harris, J. K., E. S. Haas, D. Williams, and D. N. Frank. 2001. New insight into RNase P RNA structure from comparative analysis of the archaeal RNA. RNA **7**: 220–32.

50 Higgs, P. G.. 1995. Thermodynamic properties of transfer RNA: a computational study. J. Chem. Soc. Faraday Trans. **91**, 2531–40.

51 Höchsmann, M., T. Töller, R. Giegerich, and S. Kurtz. 2003. Local similarity in RNA secondary structures. Proc. Computational Systems Bioinformatics Conf., Stanford, CA, 159–68.

52 Hofacker, I. L., S. H. F. Bernhart, and P. F. Stadler. 2004. Alignment of RNA base pairing probability matrices. Bioinformatics **20**: 2222–27.

53 Hofacker, I. L. and P. F. Stadler. 2004. The partition function variant of Sankoff's algorithm. In Proc. Int. Conf on Computational Science, Krakow, 728–35.

54 Hofacker, I. L., M. Fekete, C. Flamm, M. A. Huynen, S. Rauscher, P. E. Stolorz, and P. F. Stadler. 1998. Automatic detection of conserved RNA structure elements in complete RNA virus genomes. Nucleic Acids Res. **26**: 3825–36.

55 Hofacker, I. L., M. Fekete, and P. F. Stadler. 2002. Secondary structure prediction for aligned RNA sequences. J. Mol. Biol. **319**: 1059–66.

56 Hofacker, I. L., W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker, and P. Schuster. 1994. Fast folding and comparison of RNA secondary structures. Monatsh. Chem. **125**: 167–88.

57 HOFACKER, I. L., B. PRIWITZER, AND P. F. STADLER. 2004. Prediction of locally stable RNA secondary structures for genome-wide surveys. Bioinformatics **20**: 191–98.

58 HOFACKER, I. L., P. SCHUSTER, AND P. F. STADLER. 1998. Combinatorics of RNA secondary structures. Discr. Appl. Math. **89**: 177–207.

59 HOFACKER, I. L. AND P. F. STADLER. 1999. Automatic detection of conserved base pairing patterns in RNA virus genomes. Comp. & Chem. **23**: 401–14.

60 HOFACKER, I. L. AND P. F. STADLER. 2006. Memory efficient folding algorithms for circular RNA secondary structures. Bioinformatics **22**: 1172–6.

61 HOFACKER, I. L., R. STOCSITS, AND P. F. STADLER. 2004. Conserved RNA secondary structures in viral genomes: a survey. Bioinformatics **20**: 1495–99.

62 HOLBROOK, S. R.. 2005. RNA structure: the long and the short of it. Curr. Opin. Struct. Biol. **15**: 302–8.

63 HOLMES, I.. 2005. Accelerated probabilistic inference of RNA structure evolution. BMC Bioinformatics **6**: 73.

64 HOLMES, I. AND G. M. RUBIN. 2002. Pairwise RNA structure comparison with stochastic context-free grammars. *Pac. Symp. Biocomput.* 2002: 163–74.

65 HULL HAVGAARD, J., R. LYNGSØ, G. STORMO, AND J. GORODKIN. 2005. Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%. Bioinformatics **21**: 1815–24.

66 ISAMBERT, H. AND E. D. SIGGIA. 2000. Modeling RNA folding paths with pseudoknots: application to hepatitis delta virus ribozyme. Proc. Natl Acad. Sci. USA **97**: 6515–20.

67 JANSSON, J. AND A. LINGAS. 2003. A fast algorithm for optimal alignment between similar ordered trees. Fund. Inf. **56**: 105–20.

68 JIANG, T., G. LIN, B. MA, AND K. ZHANG. 2002. A general edit distance between beteen RNA structures. J. Comput. Biol. **9**: 371–88.

69 JIANG, T., J. WANG, AND K. ZHANG. 1995. Alignment of trees – an alternative to tree edit. Theor. Comput. Sci. **143**: 137–48.

70 JIMÉNEZ, V. M. AND A. MARZAL. 2000. Computation of the *n* best parse trees for weighted and stochastic context-free grammars. Proc. Joint Int. Workshops on Advances in Pattern Recognition, Spain: 183–92.

71 JUAN, V. AND C. WILSON. 1999. RNA secondary structure prediction based on free energy and phylogenetic analysis. J. Mol. Biol. **289**: 935–47.

72 KLEIN, P.. 1998. Computing the edit distance between unrooted ordered trees. In Proc. Annu. Eur. Symp. on Algorithms, Venice, 91–102.

73 KLEIN, R. J. AND S. R. EDDY. 2003. RSEARCH: finding homologs of single structured RNA sequences. *BMC Bioinformatics 4*(44), 1471–2105.

74 KNUDSEN, B. AND J. HEIN. 2003. Pfold: RNA secondary structure prediction using stochastic context-free grammars. Nucleic Acids Res. **31**: 3423–28.

75 KNUDSEN, B. AND J. J. HEIN. 1999. Using stochastic context free grammars and molecular evolution to predict RNA secondary structure. Bioinformatics **15**: 446–54.

76 LE, S.-Y., J.-H. CHEN, K. CURREY, AND J. MAIZEL. 1988. A program for predicting significant RNA secondary structures. CABIOS **4**: 153–59.

77 LE, S. Y. AND M. ZUKER. 1991. Predicting common foldings of homologous RNAs. J. Biomol. Struct. Dyn. **8**: 1027–44.

78 LEE, D. AND K. HAN. 2002. Prediction of RNA pseudoknots – comparative study of genetic algorithms. Genome Inf. **13**: 414–5.

79 LENHOF, H.-P., K. REINERT, AND M. VINGRON. 1998. A polyhedral approach to RNA sequence structure alignment. J. Comput. Biol. **5**: 517–30.

80 LEYDOLD, J. AND P. F. STADLER. 1998. Minimal cycle basis of outerplanar graphs. Elec. J. Comb. **5**: 209–22.

81 LIAO, B. AND T. WANG. 2002. An enumeration of RNA secondary structure. Math. Appl. **15**: 109–12.

82 LOUISE-MAY, S., P. AUFFINGER, AND E. WESTHOF. 1996. Calculations of

nucleic acid conformations. Curr. Opin. Struct. Biol. **6**: 289–98.

83 LOWE, T. M. AND S. EDDY. 1997. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. Nucleic Acids Res. **25**: 955–64.

84 LÜCK, R., S. GRÄF, AND G. STEGER. 1999. ConStruct: a tool for thermodynamic controlled prediction of conserved secondary structure. Nucl. Acids Res. **27**: 4208–17.

85 LÜCK, R., G. STEGER, AND D. RIESNER. 1996. Thermodynamic prediction of conserved secondary structure: application to the RRE element of HIV, the tRNA-like element of CMV, and the mRNA of prion protein. J. Mol. Biol. **258**: 813–26.

86 LYNGSØ, R. B. AND C. N. S. PEDERSEN. 2000. RNA pseudoknot prediction in energy-based models. J. Comput. Biol. **7**: 409–27.

87 LYNGSØ, R. B., M. ZUKER, AND C. N. PEDERSEN. 1999. Fast evaluation of internal loops in RNA secondary structure prediction. Bioinformatics **15**: 440–45.

88 MA, B., L. WANG, AND K. ZHANG. 2002. Computational similarity between RNA structures. Theor. Comput. Sci. **276**: 111–32.

89 MARTINEZ, H. M.. 1984. An RNA folding rule. Nucl. Acid Res. **12**: 323–35.

90 MATHEWS, D. H., M. D. DISNEY, J. L. CHILDS, S. J. SCHROEDER, M. ZUKER, AND D. H. TURNER. 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction ofRNA secondary structure. Proc. Natl Acad. Sci. USA **101**: 7287–92.

91 MATHEWS, D. H., J. SABINA, M. ZUKER, AND H. TURNER. 1999. Expanded sequence dependence of thermodynamic parameters provides robust prediction of RNA secondary structure. J. Mol. Biol. **288**: 911–40.

92 MATHEWS, D. H. AND D. H. TURNER. 2002. Dynalign: an algorithm for finding secondary structures common to two RNA sequences. J. Mol. Biol. **317**: 191–203.

93 MCCASKILL, J.. 1990. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. Biopolymers **29**: 1105–19.

94 MCCUTCHEON, J. P. AND S. R. EDDY. 2003. Computational identification of non-coding RNAs in *Saccharomyces cerevisiae* by comparative genomics. Nucleic Acids Res. **31**: 4119–28.

95 MIRONOV, A. A., L. P. DYAKONOVA, AND A. E. KISTER. 1985. A kinetic approach to the prediction of RNA secondary structures. J. Biomol. Struct. Dyn. **2**: 953.

96 MISSAL, K., D. ROSE, AND P. F. STADLER. 2005. Non-coding RNAs in *Ciona intestinalis*. Bioinformatics **21 (Suppl. 2)**: i77–8.

97 MÜCKSTEIN, U., I. L. HOFACKER, AND P. F. STADLER. 2002. Stochastic pairwise alignments. Bioinformatics **18**: 153–60.

98 MULLER, G., C. GASPIN, A. ETIENNE, AND E. WESTHOF. 1993. Automatic display of RNA secondary structures. Comput. Appl. Biosci. **9**: 551–61.

99 NEEDLEMAN, S. B. AND C. D. WUNSCH. 1970. A general method applicable to the search for similarities in the aminoacid sequences of two proteins. J. Mol. Biol. **48**: 443–52.

100 NOTREDAME, C., D. HIGGINS, AND J. HERINGA. 2000. T-coffee: a novel method for multiple sequence alignments. J. Mol. Biol. **302**: 205–17.

101 NUSSINOV, R., G. PIECZNIK, J. R. GRIGGS, AND D. J. KLEITMAN. 1978. Algorithms for loop matching. SIAM J. Appl. Math. **35**: 68–82.

102 PEDERSEN, J. S., I. M. MEYER, R. FORSBERG, AND J. HEIN. 2004. An evolutionary model for protein-coding regions with conserved RNA structure. Mol. Biol. Evol. **21**: 1913–22.

103 PEDERSEN, J. S., I. M. MEYER, R. FORSBERG, P. SIMMONDS, AND J. HEIN. 2004. A comparative method for finding and folding RNA secondary structures within protein-coding regions. Nucleic Acids Res. **32**: 4925–36.

104 REEDER, J. AND R. GIEGERICH. 2004. Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. BMC Bioinformatics **5**: 104.

105 REHMSMEIER, M., P. STEFFEN, M. HÖCHSMANN, AND R. GIEGERICH. 2004. Fast and effective prediction of microRNA/target duplexes. RNA **10**: 1507–17.

106 REIDYS, C., P. F. STADLER, AND P. SCHUSTER. 1997. Generic properties of combinatory maps: neutral networks of RNA secondary structures. Bull. Math. Biol. **59**: 339–97.

107 REIDYS, C. M. AND P. F. STADLER. 2002. Combinatorial landscapes. SIAM Rev. **44**: 3–54.

108 Repsilber, D., S. Wiese, M. Rachen, A. W. Schroder, D. Riesner, and G. Steger . 1999. Formation of metastable RNA structures by sequential folding during transcription: time-resolved structural analysis of potato spindle tuber viroid (−)-stranded RNA by temperature-gradient gel electrophoresis. RNA **5**: 574–84.

109 RIVAS, E. AND S. R. EDDY. 1999. A dynamic programming algorithm for RNA structure prediction including pseudoknots. J. Mol. Biol. **285**: 2053–68.

110 RIVAS, E. AND S. R. EDDY. 2001a. Noncoding RNA gene detection using comparative sequence analysis. BMC Bioinformatics **2**: 19.

111 RIVAS, E. AND S. R. EDDY. 2001b. Noncoding RNA gene detection using comparative sequence analysis. BMC Bioinformatics **2**: 8.

112 RIVAS, E., R. J. KLEIN, T. A. JONES, AND S. R. EDDY. 2001. Computational identification of non-coding RNAs in *E. coli* by comparative genomics. Curr. Biol. **11**: 1369–73.

113 RUAN, J., G. D. STORMO, AND W. ZHANG. 2004. An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots. Bioinformatics **20**: 58–66.

114 SANKOFF, D.. 1985. Simultaneous solution of the RNA folding, alignment, and proto-sequence problems. SIAM J. Appl. Math. **45**: 810–25.

115 SCHULTES, E. A. AND D. P. BARTEL. 2000. One sequence, two ribozymes: Implications for the emergence of new ribozyme folds. Science **289**: 448–52.

116 SCHUSTER, P., W. FONTANA, P. F. STADLER, AND I. L. HOFACKER. 1994. From sequences to shapes and back: a case study in RNA secondary structures. Proc. R. Soc. Lond. B **255**: 279–84.

117 SHAPIRO, B. A.. 1988. An algorithm for comparing multiple RNA secondary stuctures. CABIOS **4**: 387–93.

118 SHAPIRO, B. A., J. MAIZEL, L. E. LIPKIN, K. CURREY, AND C. WHITNEY. 1984. Generating non-overlapping displays of nucleic acid secondary structure. Nucleic Acids Res. **12**: 75–88.

119 SHAPIRO, B. A. AND K. ZHANG. 1990. Comparing multiple RNA secondary structures using tree comparisons. CABIOS **6**: 309–18.

120 SHASHA, D. AND K. ZHANG. 1990. Fast algorithm for the unit cost editing distance between trees. J. Algorithms **11**: 581–621.

121 SIEBERT, S. AND R. BACKOFEN. 2003. MARNA: a server for multiple alignment of RNAs. In Proc. German Conf. on Bioinformatics, München, 135–140.

122 STADLER, P. F. AND C. FLAMM. 2003. Barrier trees on poset-valued landscapes. Genet. Prog. Evolv. Mach. **7(20)**: 4.

123 STEGER, G., H. HOFMANN, J. FORTSCH, H. J. GROSS, J. W. RANDLES, H. L. SANGER, AND D. RIESNER. 1984. Conformational transitions in viroids and virusoids: comparison of results from energy minimization algorithm and from experimental data. J. Biomol. Struct. Dyn. **2**: 543–71.

124 SUVERNEV, A. AND P. FRANTSUZOV. 1995. Statistical description of nucleic acid secondary structure folding. J. Biomol. Struct. Dyn. **13**: 135–44.

125 TABASKA, J. E., R. B. CARY, H. N. GABOW, AND G. D. STORMO. 1998. An RNA folding method capable of identifying pseudoknots and base triples. Bioinformatics **14**(8): 691–9.

126 TACKER, M., W. FONTANA, P. F. STADLER, AND P. SCHUSTER. 1994. Statistics of RNA melting kinetics. Eur. Biophys. J. **23**: 29–38.

127 TACKER, M., P. F. STADLER, E. G. BORNBERG-BAUER, I. L. HOFACKER, AND P. SCHUSTER. 1996. Algorithm independent properties of RNA structure prediction. Eur. Biophys. J. **25**: 115–30.

128 TAI, K.. 1979. The tree-to-tree correction problem. J. ACM **26**: 422–33.

129 THIRUMALAI, D.. 1998. Native secondary structure formation in RNA may be a slave to tertiary folding. Proc. Natl Acad. Sci. USA **95**: 11506–8.

130 THIRUMALAI, D., N. LEE, S. A. WOODSON, AND D. K. KLIMOV. 2001. Early events in RNA folding. Annu. Rev. Phys. Chem. **52**: 751–62.

131 THURNER, C., C. WITWER, I. HOFACKER, AND P. F. STADLER. 2004. Conserved RNA secondary structures in Flaviviridae genomes. J. Gen. Virol. **85**: 1113–24.

132 VALIENTE, G.. 2001. An efficient bottom-up distance between trees. In Proc. Int. Symp. on String Processing and Information Retrieval, Laguna De San Raphael: 212–9.

133 VITRESCHAK, A. G., D. A. RODIONOV, A. A. MIRONOV, AND M. S. GELFAND. 2004. Riboswitches: the oldest mechanism for the regulation of gene expression? Trends Genet. **20**: 44–50.

134 VOSS, B., C. MEYER, AND R. GIEGERICH. 2004. Evaluating the predictability of conformational switching in RNA. Bioinformatics **20**: 1573–82.

135 WALTER, A., D. TURNER, J. KIM, M. LYTTLE, P. MÜLLER, D. MATHEWS, AND M. ZUKER. 1994. Coaxial stacking of helixes enhances binding of oligoribonucleotides and improves predicions of RNA folding. Proc. Natl Acad. Sci. USA **91**: 9218–22.

136 WASHIETL, S. AND I. L. HOFACKER. 2004. Consensus folding of aligned sequences as a new measure for the detection of functional RNAs by comparative genomics. J. Mol. Biol. **342**: 19–30.

137 WASHIETL, S., I. L. HOFACKER, M. LUKASSER, A. HÜTTENHOFER, AND P. F. STADLER. 2005. Mapping of conserved RNA secondary structures predicts thousands of functional non-coding RNAs in the human genome. Nat. Biotechnol. **23**: 1383–90.

138 WASHIETL, S., I. L. HOFACKER, AND P. F. STADLER. 2005. Fast and reliable prediction of noncoding RNAs. Proc. Natl Acad. Sci. USA **102**: 2454–59.

139 WATERMAN, M. S.. 1978. Secondary structure of single-stranded nucleic acids. Studies on foundations and combinatorics. Adv. Math. Supplement. Studies **1**: 167–212.

140 WATERMAN, M. S. AND T. F. SMITH. 1978. RNA secondary structure: a complete mathematical analysis. Math. Biosci. **42**: 257–66.

141 WITWER, C., I. L. HOFACKER, AND P. F. STADLER. 2004. Prediction of consensus RNA secondary structures including pseudoknots. IEEE/ACM Trans. Comput. Biol. Bioinf. **1**: 65–77.

142 WITWER, C., S. RAUSCHER, I. L. HOFACKER, AND P. F. STADLER. 2001. Conserved RNA secondary structures in picornaviridae genomes. Nucleic Acids Res. **29**: 5079–89.

143 WOLFINGER, M. T., W. A. SVRCEK-SEILER, C. FLAMM, I. L. HOFACKER, AND P. F. STADLER. 2004. Exact folding dynamics of RNA secondary structures. J. Phys. A **37**: 4731–41.

144 WUCHTY, S., W. FONTANA, I. L. HOFACKER, AND P. SCHUSTER. 1999. Complete suboptimal folding of RNA and the stability of secondary structures. Biopolymers **49**: 145–165.

145 XIA, T., J. SANATLUCIA JR., M. E. BURKARD, R. KIERZEK, S. J. SCHROEDER, X. JIAO, C. COX, AND D. H. TURNER. 1998. Parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson–Crick pairs. Biochemistry **37**: 14719–35.

146 YOUNGER, D. H.. 1967. Recognition and parsing of context-free languages in time n^3. Inf. Control **10**: 189–208.

147 ZHANG, K.. 1995. Algorithms for the constrained editing problem between ordered labeled trees and related problems. Pattern Recogn. **28**: 463–74.

148 ZHANG, K. AND D. SHASHA. 1989. Simple fast algorithms for the editing distance between trees and related problems. SIAM J. Comput. **18**: 1245–62.

149 ZUKER, M.. 1989. On finding all suboptimal foldings of an RNA molecule. Science **244**: 48–52.

150 ZUKER, M. AND P. STIEGLER. 1981. Optimal computer folding of larger RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res. **9**: 133–48.

151 ZWIEB, C., I. WOWER, AND J. WOWER. 1999. Comparative sequence analysis of tmRNA. Nucleic Acids Res. **27**: 2063–71.

15
RNA Tertiary Structure Prediction

François Major and Philippe Thibault

1 Introduction

During the last decade, the number of high-quality X-ray crystallographic RNA three-dimensional (3-D) structures has increased significantly, and the resolution of the large ribosomal subunit crystal structure was considered a major step towards a better understanding of RNA tertiary structure and folding. The recent discovery of the RNA interference (RNAi) pathway (see Chapter 45) has also contributed greatly to the popularity of RNAs, by suggesting their direct implication in genetic expression and regulation. More than ever, determining rapidly and precisely the tertiary structure and function of noncoding RNAs is a crucial step towards our understanding of several cellular metabolic pathways.

This chapter is dedicated to the RNA tertiary structure prediction problem – the determination of the complete set of chemical interactions (and therefore 3-D fold) of an RNA from sequence data. To achieve RNA structure prediction, one needs to discover and apply its structural and architectural principles, which can be learnt from thermodynamics, as well as from structural data gathered from X-ray crystallography and other high-resolution, but also low-resolution, experimental methods. Here, we present a series of nomenclatures and formalisms to describe RNA tertiary structure, as well as computer data structures and algorithms that implement three important research activities with the aim of solving RNA tertiary structure prediction: annotation, motif discovery and modeling.

We present in Section 2 a series of RNA structure components and the terms employed by the RNA specialists to discuss them – their universe of discourse (nowadays referred to as their ontology). First, we present an ontology of nucleotide conformations and binary interactions. Then, visual or automated inspection of RNA 3-D structures is necessary to depict higher-order architectural principles (the next abstraction levels). In Section 3, we introduce a definition of *n*-ary nucleotide interactions to describe RNA higher-order motifs, which are found repeated in RNA structures, and are often

Bioinformatics - From Genomes to Therapies Vol. 1. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

linked to specific structural and biochemical functions, and an approach to search them. Finally, in Section 4, we present how accurate computer models of RNA tertiary structures can be generated and how, by challenging them experimentally, they bring insights about function.

The flowchart in Figure 1 shows the relationships between structural data and hypotheses, and how the research activities that aim at solving the RNA tertiary structure prediction problem are intimately linked. The high-resolution (better than 3 Å) X-ray crystal structures of the Protein Data Bank (PDB) (www.rcsb.org) [1] constitute an excellent structural (learning) data set that aids the research, and from which RNA tertiary structure prediction algorithms can be inspired and tested. The characterization and formalization of RNA structural data (annotation), the discovery of high-order components (motif discovery) and the building of RNA tertiary structure models (modeling) contribute directly to the learning and discovery processes, leading to new knowledge that is fed back to research.

An ultimate solution to the tertiary structure prediction problem will provide us with invaluable structural information, and will allow us to determine the function and the evolutionary relationships of RNAs. Knowledge of RNA tertiary structure impacts on molecular medicine techniques to control genetic expression, and to inhibit and activate specific cellular functions. The cell controls its own genetic expression by processing micro RNAs through the RNAi pathway. As we discover and characterize the elements of RNAi, we learn how to design RNAs that interfere and block the expression of several genes. Knowledge of the structure and of the interplay between the RNAs and the other RNAi elements is fundamental. Alternatively, we could target the natural micro RNAs of the cell using drugs. Again, knowledge of the targeted RNA structure is necessary to design accurate drugs. Targeting the noncoding RNAs of the cell allows us to manipulate its fundamental mechanisms prior to protein translation; like playing with the "source code" of the cell. Antibiotics such as aminoglycosides and macrolides target the site-A of prokaryotic ribosomes, blocking protein translation. The search and discovery of other sites in the ribosome or in other RNAs involved in such fundamental mechanisms require the determination of their tertiary structure if we want to design drugs capable of inhibiting them. Ribozymes are catalytic RNAs that can cleave a substrate efficiently and precisely. For instance, ribozymes can be used to cleave a messenger prior to its translation by the ribosome. Here, again, knowledge of the tertiary structure of ribozymes and of their complex with the substrate is essential for rational design.

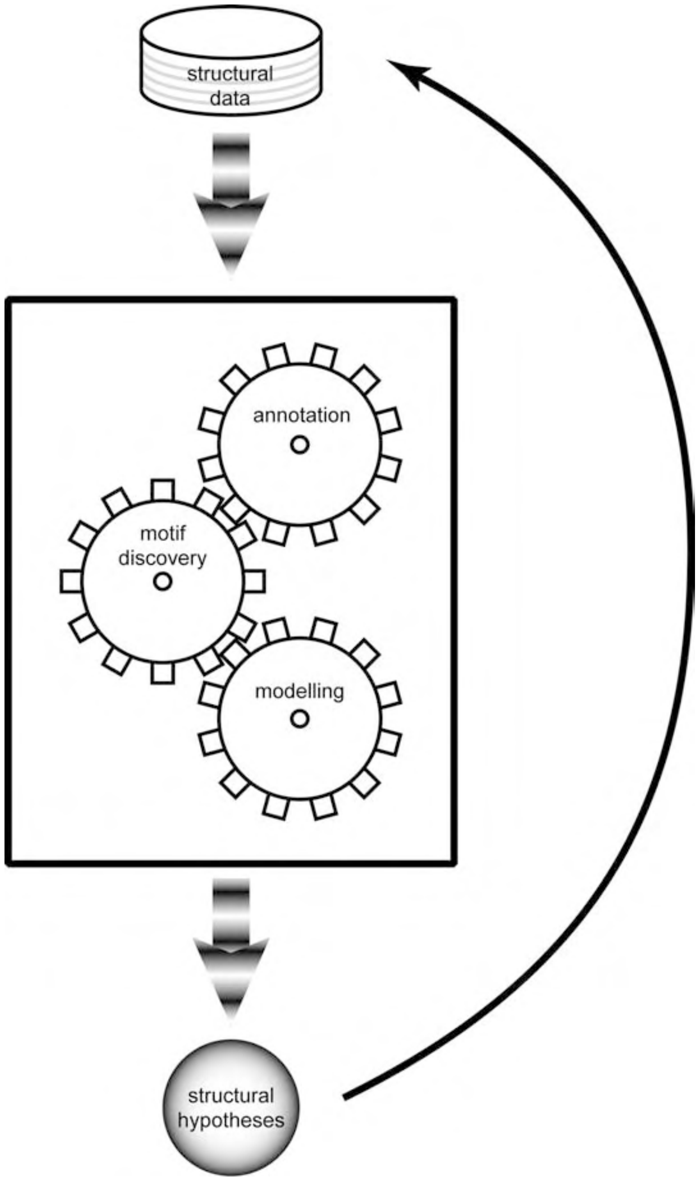


Figure 1 Data flow and research activities of tertiary structure prediction. Structural data are used to develop the (computational) tools employed by the researchers to annotate, inspect and model RNA tertiary structures. Structural hypotheses are generated and, when challenged experimentally, bring new structural data to research.

2 Annotation

The annotation of an RNA tertiary structure is the assignment (manual or automated) of appropriate symbols, taken from the RNA ontology, that apply to a given RNA. One can see annotation as a data refinement process that complements the 3-D atomic coordinates – a different and perhaps higher level of abstraction which can be thought of as an efficient and sound data format to study further tertiary structures.

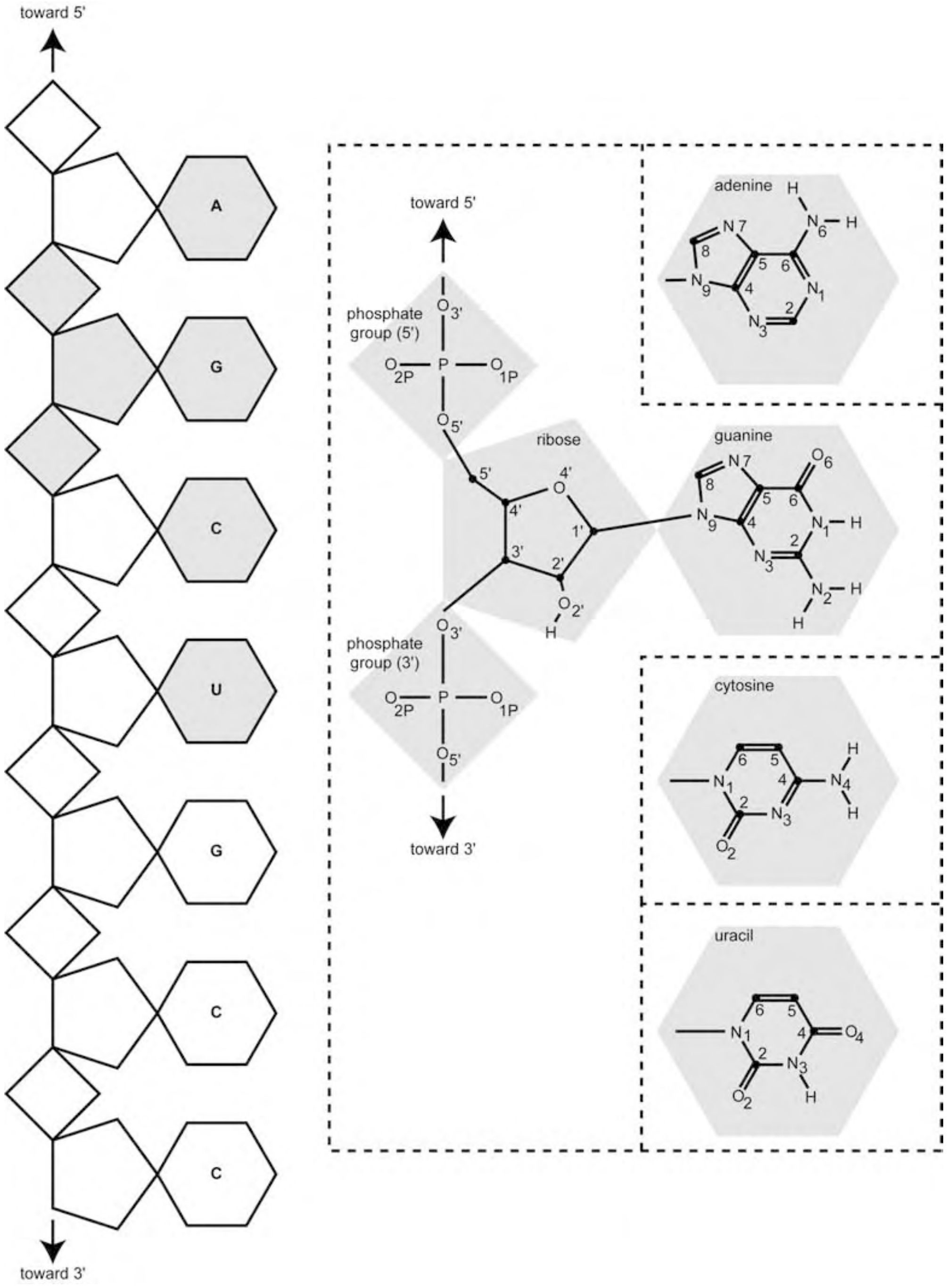
A human RNA expert recognizes the attributes of tertiary structures by visualization using interactive computer graphics and can therefore annotate a given RNA 3-D structure. An automated procedure loads the RNA 3-D atomic coordinates in memory and then computes the annotations. Gendron and coworkers have developed a computer program, MC-Annotate, which annotates a fraction of the current RNA ontology, in particular the terms related to the nucleotide conformations, as well as base stacking and base pairing types [2] (see Section 2.2). MC-Annotate can be run over the web (www-lbit.iro.umontreal.ca; under Research and MC-Annotate). Westhof and coworkers, in collaboration with the PDB, developed RNAView, a computer program that draws the secondary structure of an RNA while using the *LW* nomenclature (see Sections 2.2.2) to display the base pair types [3]. RNAView is accessible on the web (ndbserver.rutgers.edu/services).

In this section, we present a series of RNA tertiary structure attributes and how they can be computed from 3-D atomic coordinates from X-ray crystallographic structures of the PDB. We present the nucleotide conformations (Section 2.1) and interactions (Section 2.2) that are needed to define higher-order RNA components (Section 3), and to build and describe RNA tertiary structure (Section 4).

2.1 Nucleotide Conformations

RNAs are polymeric molecules. The monomer unit is a ribonucleotide, or simply nucleotide, which divides in three units: the nucleobase (or simply base), the ribose and the phosphate group (see Figure 2). There are four bases: adenine (A), guanine (G), cytosine (C) and uracil (U). The four bases partition in two families: the pyrimidines (Y) C and U, which are composed of a single pyrimidine ring, and the purines (R) A and G, which are composed of the fusion of the pyrimidine ring ($C_4H_4N_2$) and an imidazole ring ($C_3H_4N_2$). The

Figure 2 RNA chemical structure. The polynucleotide chain (on the left) is made of the bases (hexagons), riboses (pentagons) and phosphate groups (diamonds). The four common types of bases (on the right): the two purines adenine and guanine, and the two pyrimidines cytosine and uracil. The phosphodiester linkage (middle) connects two nucleotides. The ribose (center) links two phosphate groups: one to its 5′ oxygen (above) and the other to its 3′ oxygen (below). The conventional atomic numbering system is used. Small black circles represent the carbon atoms and their complementary hydrogen atoms are not shown.


toward 5'
A
G
C
U
G
C
C
toward 3'
toward 5'
phosphate group (5')
O3'
O2P
P
O1P
O5'
ribose
5'
4'
O
4'
1'
3'
2'
O2'
H
phosphate group (3')
O3'
O2P
P
O1P
O5'
toward 3'
adenine
H
N6
H
N 7
8
5
6
N9
4
N1
2
N3
guanine
N 7
O6
8
5
6
N9
4
N1
H
2
N3
N2
H
H
cytosine
H
6
5
N1
4
N4
2
N3
H
O2
uracil
6
5
N1
4
O4
2
N3
O2
H

International Union of Pure and Applied Chemistry (IUPAC) defined a one-letter code for all possible subsets of {A, C, G, U} (shown in Table 1).

Table 1 IUPAC nucleotide nomenclature

Code	Nucleotide subset
M	{A, C}
R	{A, G}
W	{A, U}
S	{C, G}
Y	{C, U}
K	{G, U}
V	{A, C, G}
H	{A, C, U}
D	{A, G, U}
B	{C, G, U}
N	{A, C, G, U}

The ribose links the phosphate groups to which the bases are attached by the glycosidic bond: C1′–N9 in purines and C1′–N1 in pyrimidines. The riboses and the phosphate groups constitute the backbone, and are linked through diester bonds: C5′–O5′ and C3′–O3′. The chain C3′–O3′–P–O5′–C5′ from one ribose to another is referred to as the phosphodiester linkage that ties the nucleotides together (see Figure 2).

When 3-D points represent the center of the atoms in the structure, the covalent bond lengths, and the bond and torsion angles can be computed directly. The covalent bond length between atoms A and B is simply defined by the Cartesian distance between points A and B. The covalent bond angle between atoms A, B and C is defined by the angle between vectors $\overrightarrow{BA}$ and $\overrightarrow{BC}$. Finally, the covalent bond torsion angle between atoms A, B, C and D is defined by the angle between the projection of $\overrightarrow{BA}$ and $\overrightarrow{CD}$ in a plane perpendicular to $\overrightarrow{CD}$. In general, bond lengths and angles are considered constant in most computer prediction systems. Consequently, the 3-D conformation of a nucleotide can be described by its torsion angles. The phosphodiester linkage has six torsions (α, β, γ, δ, ε and ζ), the ribose has five torsions (θ_0–θ_4) and there is one torsion around the glycosidic bond (χ) (see Figure 3a).

Note the δ and θ_1 torsions are measured on the same covalent bond, C3′–C4′, but from different end-points, respectively, C2′ and O4′ in the ribose for θ_1, and C5′ and O3′ in the phosphodiester chain for δ. The glycosidic torsion, χ, is measured respectively in purines and pyrimidines from atoms O4′–C1′–N9–C4 and O4′–C1′–N1–C2. The furanose ring stereochemistry imposes interdependent relations on θ_{0-4}, which is expressed by the cosine function:

$$\theta_j = \theta_{\max} \cos(\rho + j\varphi), \tag{1}$$

where $j = 0, \ldots, 4$ and $\varphi = 144°$ (720°/5).

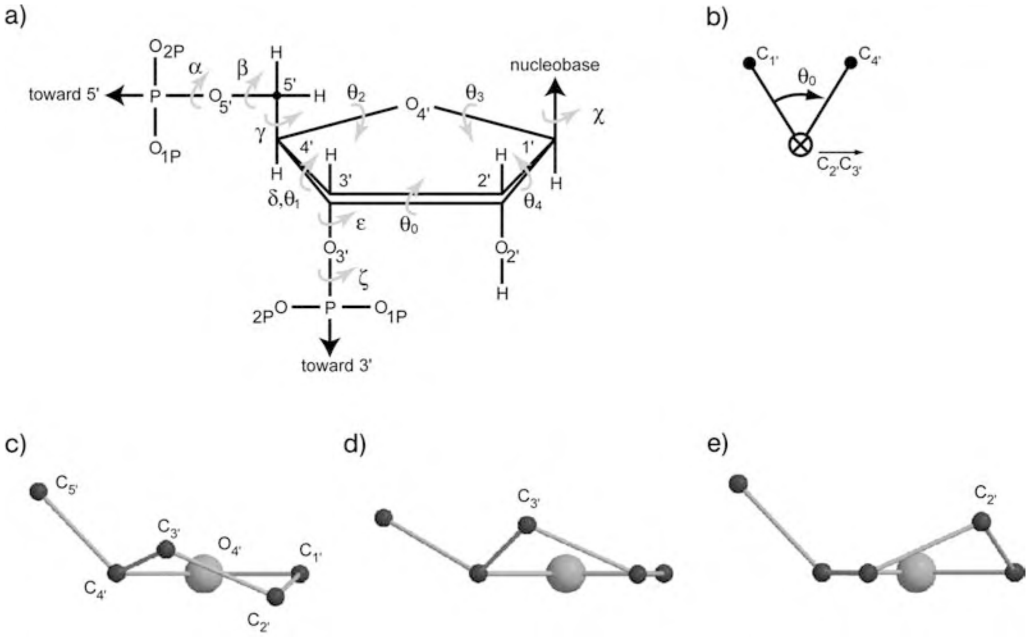


Figure 3 The nucleotide torsion angles. (a) Individual torsions are shown using grey arrows: χ on the glycosidic bond between the base and the ribose, θ_{0-4} around the ribose, and α, β, γ, δ, ε and ζ along the phosphodiester chain (θ_1 and δ are defined on the same covalent bond). (b) θ_0 measurement. The torsion is computed as the angle between the projection of vectors $\overrightarrow{C_{2'}C_{1'}}$ and $\overrightarrow{C_{3'}C_{4'}}$ in the plane perpendicular to vector $\overrightarrow{C_{2'}C_{3'}}$ (crossed circle). (c) The $^2T_3$ twist shape C2′-*exo*–C3′-*endo* ribose pucker mode. (d) The ^3E envelope shape C3′-*endo* ribose pucker mode. (e) The ^2E envelope shape C2′-*endo* ribose pucker mode.

When $j = 0$, we have:

$$\theta_0 = \theta_{max} \cos(\rho). \tag{2}$$

In Eqs. (1) and (2), ρ is the pseudorotation of the ribose ring [4]. By varying ρ from 0 to 360° by steps of 90°, θ_0 goes from θ_{max} to 0, to θ–θ_{max}, back to 0 and, finally, back to θ_{max}. The θ_{max} value is reached twice – at the initial conformation [$\rho = 0$, as $\cos(0) = 1$] and at $\rho = 360°$. At each step of $\rho + 180°$, the sign of all torsions is inversed, corresponding to the mirror image of the conformation at ρ. A useful equation is derived from Eq. (1), which determines ρ:

$$\tan\rho = \frac{(\theta_2 + \theta_4) - (\theta_1 + \theta_3)}{2\theta_0(\sin 36° + \sin 72°)}\ . \tag{3}$$

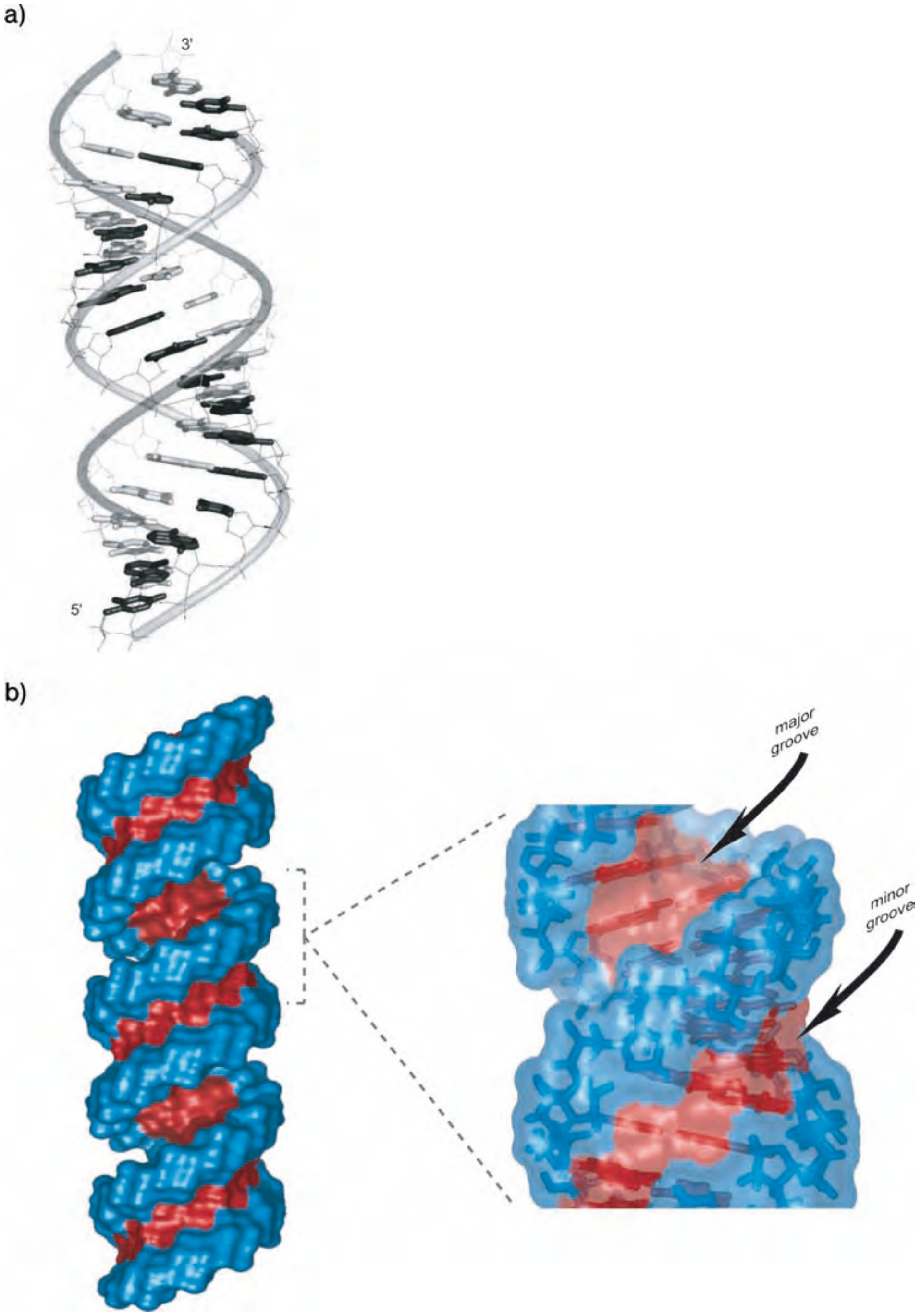
Equation (2) determines θ_{max}.

Two geometric shapes characterize the stereochemistry of the ribose: envelope and twist (or half-chair). The ribose forms an envelope when only one of the five atoms of the furanose (C1′, C2′, C3′, C4′ or O4′) is out of the plane formed by the four others. The ribose forms a twist when two atoms are out of the plane formed by the remaining three. In a 360° period of the pseudorotation angle, the ribose stereochemistry alternates from the envelope to the twist shapes, successively, on each atom. At $\rho = 0°$, the ribose is in the twist shape with the C3′ atom above and the C2′ atom below the plane, which is conveniently denoted by $^3T_2$ (numbered atom above the plane in superscript and below the plane in subscript), as illustrated in Figure 3(c). All molecule 3-D rendered images were generated using MolScript [5] and Raster3D [6], as well as PyMOL (www.pymol.org). When the ball-and-stick representation is used, sphere radii are proportional to atomic masses ($C < N < O < P$). Then, the geometry of the ribose shifts at each 18°; for $\rho \in [18°, 36°]$, the ribose forms an envelope with the C3′ atom above the plane, ^3E (see Figure 3d); for $\rho \in [36°, 54°]$, the geometry changes to $^3T_4$; $_4E$ for $\rho \in [54°, 72°]$ and so forth for the 20 different geometries [7]. The ribose geometries are referred to as the sugar pucker modes. Another widely used ribose pucker mode nomenclature among RNA structure specialists is one where the atom(s) bulging out of the plane are suffixed with either *endo* or *exo*, respectively, for above and below the plane. Thus, in the example of Figure 3(c), the $^3T_2$ shape is equally named C2′-*exo*-C3′-*endo*; C3′-*endo* for the example in Figure 3(d) and C2′-*endo* for the example in Figure 3(e).

Single-stranded RNAs fold back on themselves to form double-stranded helices in the A-RNA conformation which is similar to the A-DNA double helix. Among all 20 ribose pucker modes, the C3′-*endo* is the most common, as it is the conformation of the riboses in the Watson–Crick base pairs of the A-RNA double helix (Figure 4). The asymmetry of the Watson–Crick base pair geometry (Section 2.2.2) induces the formation of two grooves in the helix. The major groove of the A-RNA double helix is narrow and deep, whereas the minor groove is broad and shallow (Figure 4b). Theoretically, the C2′-*endo* mode, adopted by the nucleotides in the B-RNA double-helical form, is unstable because of the proximity of the 2′-OH groups to the bases. Nevertheless, a good fraction of RNAs contain nucleotides in the C2′-*endo* conformations, as in loop regions and at the extremities of a double helix.

→

Figure 4 Type A-RNA double helix. (a) The bases form stacked Watson–Crick base pairs. The 5′-strand is shown in dark; the 3′-strand in light. The thread follows the phosphorus atoms. The hydrogen atoms are not shown. (b) Major and minor grooves. The bases are in red; the backbone in blue. The 3-D structures were generated by MC-Sym (see Section 4.2).


a)
3'
5'
b)
major groove
minor groove

In addition to the pucker mode, the glycosidic torsion χ is also divided in a range of values. The *anti* conformation characterizes a base oriented away from the ribose. From χ = 180°, where the plane of the base is aligned with the O4′–C1′ bond in a direction away from O4′, the *anti* conformation covers a rotation of ±90°: $\chi \in [-180°, -90°]$ and $\chi \in [90°, 180°]$ (see Figure 5a). At χ = 0°, where the plane of the base is aligned with the O4′–C1′ bond in a direction towards the O4′ atom, the *syn* conformation covers the remaining rotations of ±90°: $\chi \in [-90°, 90°]$ (see Figure 5b). The nucleotide conformations in both the A-RNA and B-RNA double-helical forms adopt the *anti* conformation.

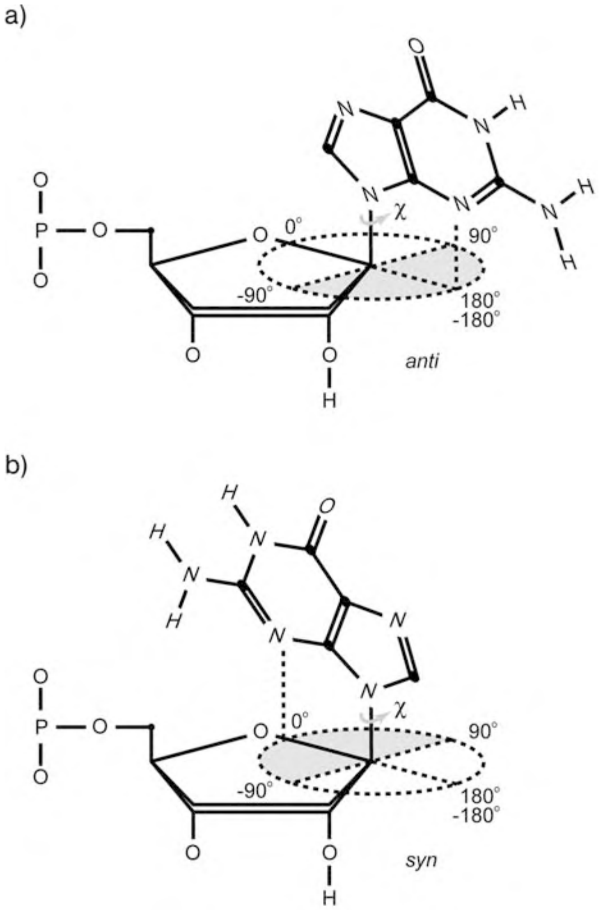


Figure 5 The glycosidic bond torsion. The ranges of χ values are shown in grey. (a) The *anti* conformation aligns the base away from the ribose. (b) The *syn* conformation aligns the base towards the ribose.

The nucleotide conformations can be annotated directly from their torsion angles: θ_{0-4} to determine the ribose puckering modes (C3′-*endo*, C2′-*endo*, etc.) and χ to determine the base to ribose relative orientation (*anti* or *syn*).

As mentioned above, the nucleotide conformation is mainly, if not completely, characterized by its free torsion angles. Consequently, many attempts aim at classifying nucleotide conformations according to torsion angles. In the late 1970s, Olson reduced the six phosphodiester chain torsions to two pseudotorsions of virtual bonds spanning the chain in two C–C–O–P segments (C4′–C5′–O5′–P and C4′–C3′–O3′–P). She reported a statistical correlation between the individual torsions and those spanned by the two pseudotorsions [8]. Gautheret and coworkers proposed a different approach that analyzed the clustering of dinucleotide conformations. They used a RMSD distance to compare pairs of dinucleotides aligned by their P–O3′ bond. The clustering discriminated families of dinucleotides with similar P–P orientations, and was used as the basis of a conformational search space in early versions of the MC-Sym computer program (see Section 4.2) [9, 10]. They observed with this approach the "crankshaft effect" [11], as different torsion patterns lead to similar 3-D conformations. In more recent studies, Duarte and coworkers extended Olson's work by defining two pseudotorsions and identified recurrent torsion patterns as well [12].

Hershkovitz and coworkers analyzed individual Gaussian distribution data fitting of the four backbone torsions α, γ, δ and ζ in the crystal structure of *Haloarcula marismortui* 23S rRNA. They identified 37 different conformers from which they defined nucleotide signatures [13]. Others, such as Murray and her colleagues, have classified three-torsion patterns (α, β, γ) and (δ, ε, ζ) into 42 conformers by applying quality filtering to high-resolution X-ray crystal structures [14]. According to them, each nucleotide conformer represents a high-quality reference nucleotide conformation. Schneider and coworkers have analyzed the torsion angles of dinucleotides by Fourier averaging of six selected 3-D distributions. They found 18 conformers, apart from the overrepresented A-RNA helical conformation [15]. Similarly to Gautheret and coworkers, they concluded the structural conformational space of RNA 3-D structures could be sampled by a small number of dinucleotide conformers.

2.2 Nucleotide Interactions

Inter-nucleotide interactions contribute to the overall stability of RNA tertiary structure. The obvious example is the stacked Watson–Crick base pairs that forms the A-RNA double helix. Interactions outside double helices that are distant in sequence are often referred to as tertiary interactions and play a major role in RNA folding. Here, we define and present a nomenclature to describe base stacking and base pairing information.

2.2.1 Base Stacking

Base stacking involves London dispersion inter-molecular interactions between two bases that induce a 3-D arrangement where one base is stacked on top of the other (see Figure 4). Bases can stack towards each side and therefore there are four different base stacking types. To identify on which side a base is stacked, a vector normal to the plane of the base is defined so that any base in a classical A-form helix have their normal vectors oriented in the same direction; towards the 3′-strand end-point. In pyrimidines, this normal vector is the rotational vector $\overrightarrow{n}_{\mathrm{Y}}$ obtained by a right-handed rotation from N1 to N6 around the pyrimidine ring. The pyrimidine ring in purines is reversed with respect to that of pyrimidines, as stacked in the A-form helix, and therefore the pyrimidine ring normal vector for purines must be reversed. We define $\overrightarrow{\sigma}$ as the normal vector for any base: $\overrightarrow{\sigma} = \overrightarrow{n}_{\mathrm{Y}}$ in cytosine and uracil, whereas $\overrightarrow{\sigma} = -\overrightarrow{n}_{\mathrm{Y}}$ in adenine and guanine (see Figure 6a). When bases A and B stack, $\overrightarrow{\sigma}_{\mathrm{A}}$ is in the same or opposite direction to $\overrightarrow{\sigma}_{\mathrm{B}}$, and B is either above or below A. Therefore, a base stack is "straight" or "reverse" and the second base is either "above" or "below".

The four cases are shown in Figure 6(b). The "upward" stacking corresponds to "straight" and "above"; "downward" to "straight" and "below"; "inward" to "reverse" and "above"; and "outward" to "reverse" and "below". Consequently, the base stack in A-form helices is "upward". The four cases can be written using the less than (<) and greater than (>) characters. For instance, if A and B stack inward, then we can simply write "A >< B" (see Figure 6b).

Note that base stacking is independent of the backbone direction. Two adjacent bases in a sequence can be stacked in any of the four cases. As an example, consider the A-riboswitch aptamer module adenine-sensing messenger RNA (mRNA) crystal structure from *Vibrio vulnificus* (PDB ID 1Y26), where U22 and A23 are stacked downward (U22 << A23) (see Figure 7). This particular stacking interaction occurs at a junction that connects two fragments inside the adenine-sensing pocket. Both U22 and A23 participate in base triples (a base simultaneously pairs to two other bases) [16].

MC-Annotate implements base stacking as in Gabb and coauthors [17], by using the distance between the ring centers, and the dihedral angle and horizontal shift between the rings. As purines are made of two rings, the pyrimidine and imidazole, both are verified. The base stacking interactions are labeled according to the nomenclature above. Note that biased cutoffs on each parameter are needed to decide whether two bases stack.

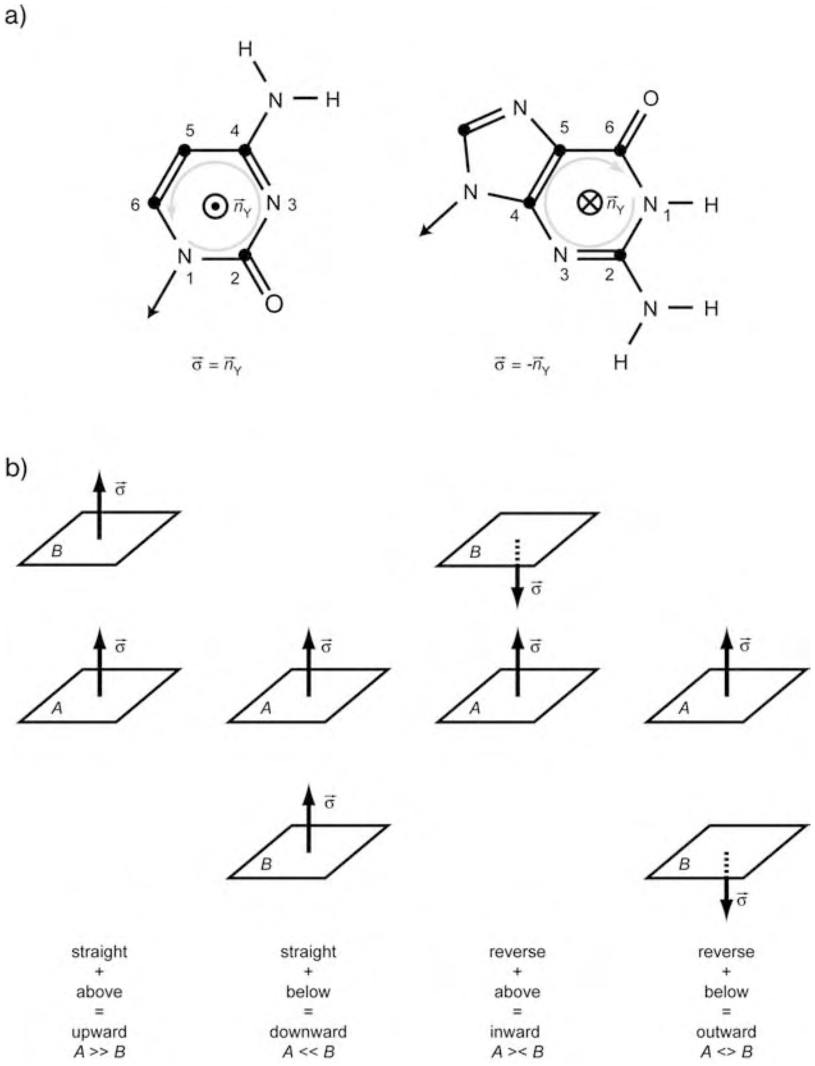

a)
H
N — H
5
4
6
N 3
$\vec{n}_Y$
N
1
2
O
$\vec{\sigma} = \vec{n}_Y$
N
O
5
6
N
$\vec{n}_Y$
N 1 — H
4
N
3
2
N — H
H
$\vec{\sigma} = -\vec{n}_Y$
b)
$\vec{\sigma}$
B
A
straight + above = upward
A >> B
straight + below = downward
A << B
reverse + above = inward
A >< B
reverse + below = outward
A <> B

Figure 6 Base stacking. (a) The base normal vector $\overrightarrow{\sigma}$ in terms of the pyrimidine ring normal vector $\overrightarrow{n}_{\mathrm{Y}}$. In pyrimidines (left, here a cytosine), $\overrightarrow{\sigma}$ is defined as $\overrightarrow{n}_{\mathrm{Y}}$, the rotational vector obtained from a right-handed rotation around the pyrimidine ring from atoms 1–6. In purines (right, here a guanine), $\overrightarrow{\sigma}$ is defined as $-\overrightarrow{n}_{\mathrm{Y}}$. (b) Nomenclature of the four stacking cases. Bases A and B are represented by planes.

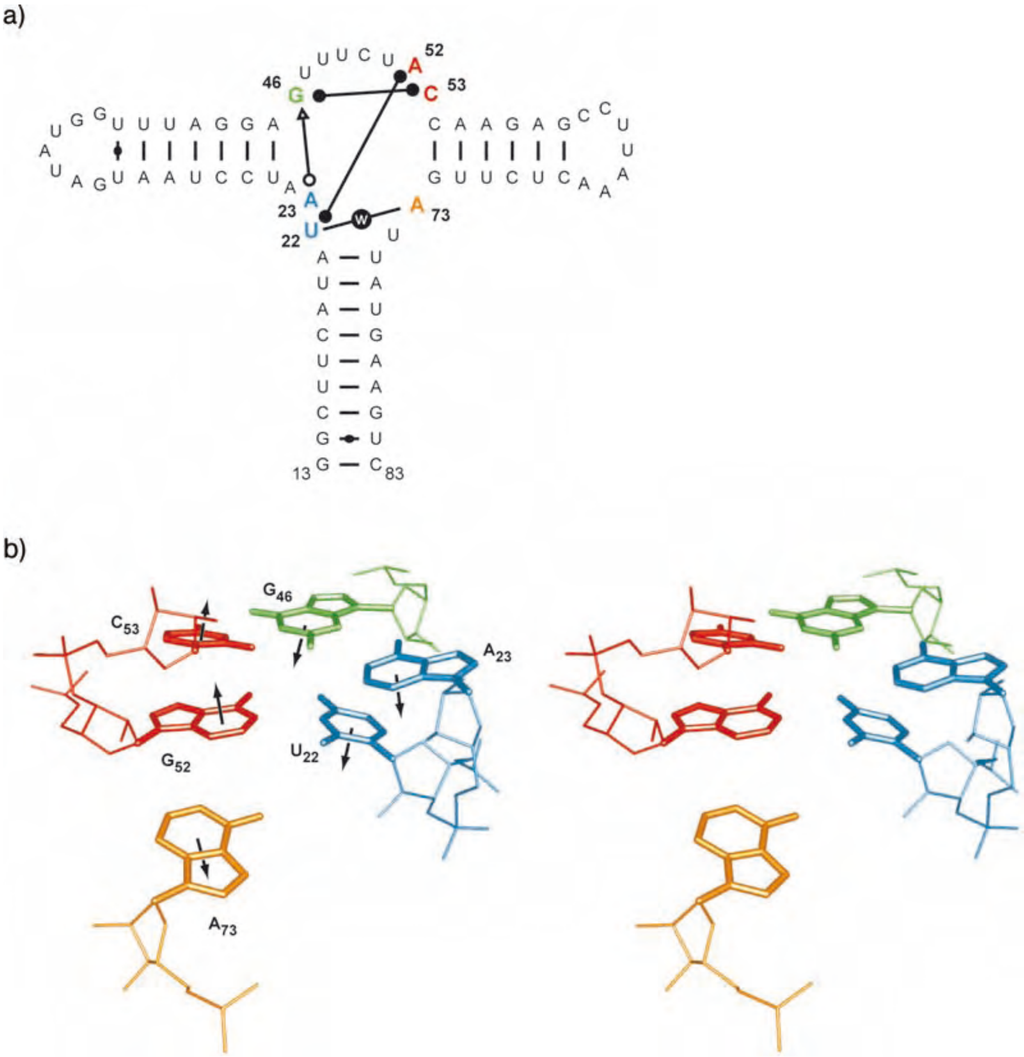


Figure 7 A-riboswitch aptamer. (a) Secondary structure. The bases shown in color are involved in two base triples: U22 (blue) with A52 (red) and A73 (orange), and G46 (green) with U23 (blue) and C53 (red). U22 and A23 are stacked downward. The *LW* nomenclature is used: w for water-mediated; circle for W; square for H; triangle for S. (b) Tertiary structure of the two base triples. The arrows indicate each normal vector, $\overrightarrow{\sigma}$

2.2.2 **Base Pairing**

Base pairing involves the formation of hydrogen bonds between exocyclic hydrogen donor groups (mainly NH and NH_2) and acceptor groups (mainly CO and N). The well-known canonical Watson–Crick G=C and A–U base pairs have three and two hydrogen bonds, respectively. Successive Watson–Crick base pairs that stack upward result in the A-form helix, also called stem (see Figure 4). The determination of the helices of an RNA from sequence data is the goal of secondary structure prediction (see Chapter 14). As stems have a tight and local 3-D structure, they are often manipulated as rigid objects in computer modeling. Other than Watson–Crick base pairs abound in RNAs; near 20% in the yeast phenylalanine transfer RNA (tRNA-Phe) and near 50% in the large ribosomal subunit), and are often qualified as "non-canonical", or non-Watson–Crick.

In his famous 1984 book, Saenger compiled 28 base pairing patterns involving at least two hydrogen bonds [7]. Each base pair was assigned a roman number: for instance XIX for the G=C Watson–Crick base pair, XX for the A–U Watson–Crick base pair, XXIII for the "Hoogsteen" A–U base pair, XXIV for the "reverse Hoogsteen" A–U base pair and XI for the "sheared" G–A base pair; see Ref. [7], p. 120 for the complete list).

More recently, Leontis and Westhof proposed a new nomenclature, *LW* [18]. In *LW*, three hydrogen bond contact edges (*W* = Watson–Crick, *H* = Hoogsteen and *S* = Sugar) were defined in each base (see Figure 8). To describe a base pair, one has simply to name its interacting edges. For the Watson–Crick base pair, since the hydrogen bonds are formed by chemical groups on the *W* edges of each base, we refer to it as W/W. In addition, the relative orientation of the riboses with respect to the plane of the base pair is annotated as *cis* or *trans*, respectively, if the glycosidic bonds extend towards the ribose are in the same (as in the A-form helix) or opposite orientation (see Figure 9a).

To introduce more precision and distinguished among possible ambiguities of the *LW*, and in particular in one-hydrogen-bond base pairs, Lemieux and Major divided each contact edge in three regions they named faces [19]. In this extension of *LW*, *LW+*, each possible hydrogen bond face is named by its corresponding *LW* edge, to which one of three possible orientations was added: w, h and s. For instance, the *W* edge has the *Ww* face at the center of the edge, the *Wh* face towards the *H* edge and the *Ws* face towards the *S* edge (see Figure 8). The wobble GU base pair is annotated W/W in *LW*, and more precisely *Ww/Ws* in *LW+*. Bifurcated hydrogen bonds that oscillate between two *LW* edges have their own faces in *LW+*: *Bh* between *W* and *H* edges, and *Bs* between *W* and *S* edges.

Finally, the normal vector $\overrightarrow{\sigma}$ used to annotate base stacking can also be used in base pairing to address the relative orientation (parallel or antiparallel) of

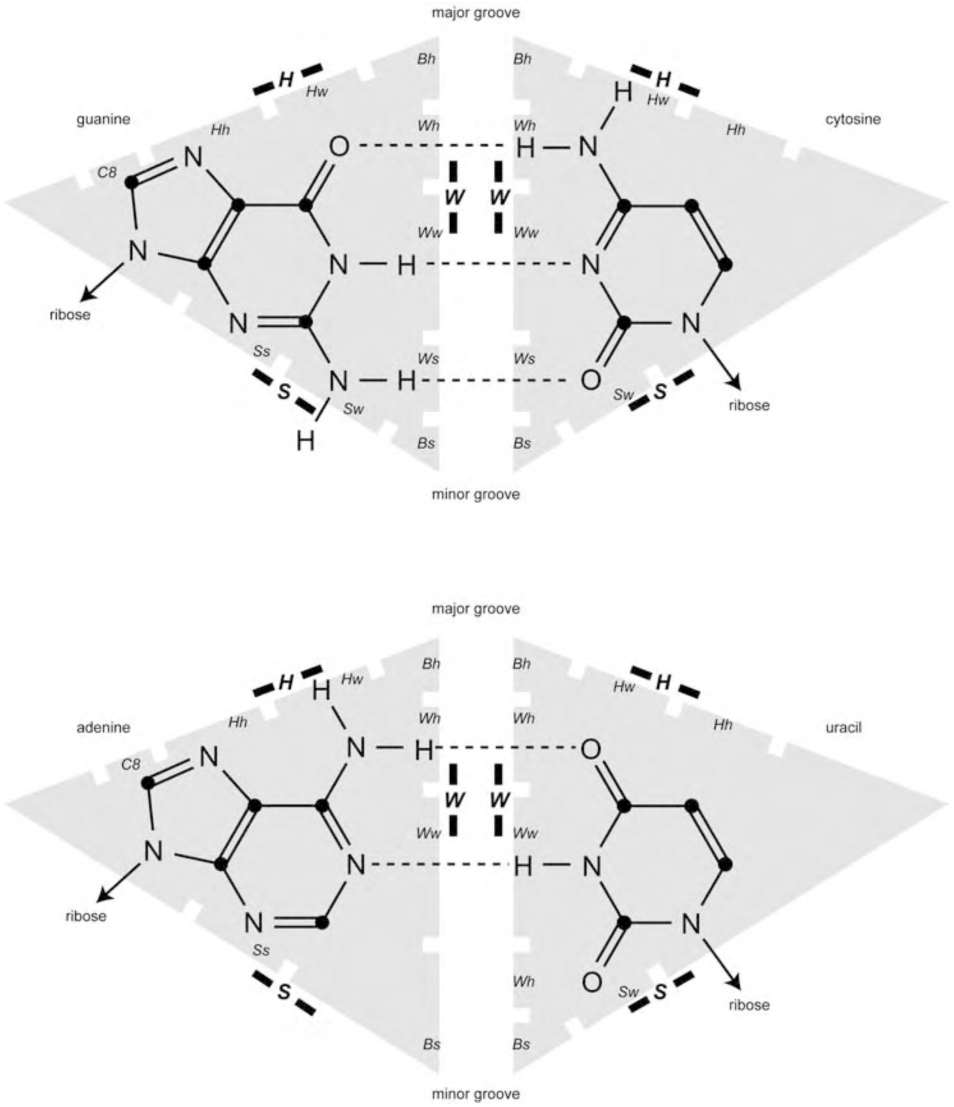


Figure 8 Base pairing patterns. The two canonical Watson–Crick base pairs and their hydrogen bonds (dashed lines) are shown (top: G=C; bottom: A–U). The *LW* nomenclature is shown by engulfing shaded triangles, where the arcs represent the W, H and S edges on the four standard bases. Here, the W edges are in contact in both base pairs. Notches along each edge delimit the *LW+* faces for each base. The major groove of the A-RNA double helix is on the H side, whereas the minor groove is on the S side.

the two bases in a base pair. The *cis* W/W base pairs in the A-form helix are characterized by the antiparallel orientation (see Figure 9b).

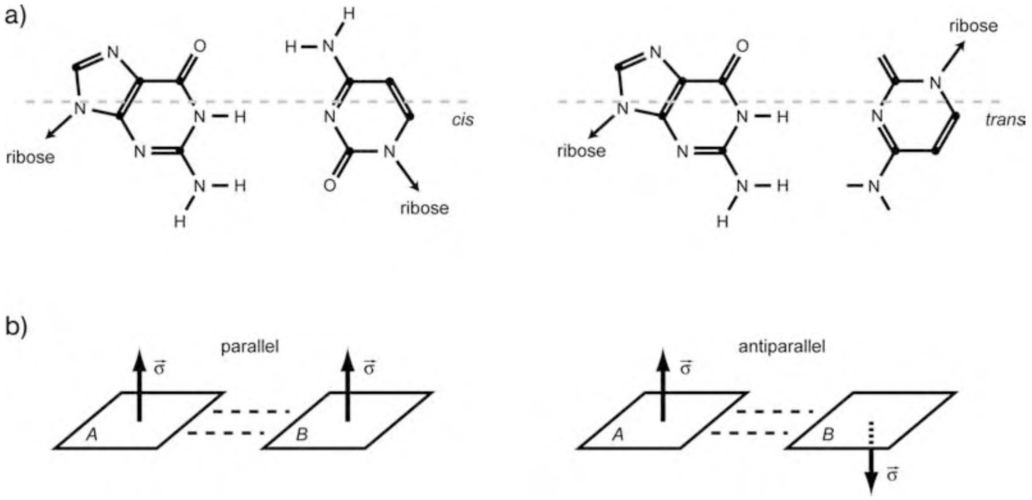


Figure 9 Base pairing orientations. (a) From the plane of the base pair, if both glycosidic bonds are oriented on the same side of the line that splits the plane evenly in two (dashed gray line), the base pair is *cis* (left). Else, if the glycosidic bonds are on each side, the base pair is *trans* (right). (b) Base normal vector $\vec{\sigma}$ relative orientation. Two bases, A and B, represented here by planes in perspective view, are either "parallel" if their $\vec{\sigma}$ are oriented in the same direction (left) or "antiparallel" if their $\vec{\sigma}$ are oriented in opposite directions (right).

In order to limit the bias of using cutoffs, Gendron and coworkers, in MC-Annotate, implemented the detection of the base pair types by using unsupervised learning [19]. Single hydrogen bonds are selected by Gaussian distribution fitting of three geometric parameters calculated between all pairs of bases: the hydrogen, the donor, the acceptor and the lone electron pair (positioned 1 Å away from the acceptor in the direction of the orbital). The subset of hydrogen bonds between two bases is selected by solving the equilibrium state of the maximal flow of the directed bipartite graph formed by all possible hydrogen bonds between the two bases. The base pair is labeled according to the *LW*+ nomenclature. The relative orientations of the glycosidic bonds (*cis* or *trans*) and of the normal vectors (parallel or antiparallel) are also computed. The RNAView annotation procedure of base pairs differs considerably from the one implemented in MC-Annotate, as it is based on geometrical cutoffs.

Lee and Gutell proposed an alternative topological nomenclature, *LG*. Starting with the Watson–Crick C=G or A–U, or even with the wobble G–U base pair, they defined 14 families by successively manipulating the base plane and glycosidic bond relative orientations: shearing, flipping, reversing, paralleling or slipping. The resulting 14 families are the Watson–Crick (*WC*), wobble (*Wb*), slipped Watson–Crick (*sWC*), slipped wobble (*sWb*), reversed Watson–Crick (*rWC*), reversed wobble (*rWb*), Hoogsteen (*H*), reversed Hoogsteen (*rH*), sheared (S), reversed sheared (*rS*), flipped sheared (*fS*), parallel flipped sheared (*pfS*), parallel sheared (*pS*) and reversed parallel sheared (*rpS*) [20].

2.2.3 Isosteric Base Pairs

Base pairs are isosteric if they preserve a local tertiary structure and, thus, function, as observed in evolutionary related RNAs whose sequences may diverge. Leontis and coworkers have superimposed the geometry of all possible base pairs according to their C1′–C1′ distances and *cis*/*trans* base orientations [21]. Then, they mapped all 16 combinations to the 12 families of the *LW* nomenclature, which resulted in isostericity matrices from which it can be shown that all canonical Watson–Crick combinations are isosteric and that the wobble G–U base pair is isosteric to the protonated A–C base pair. Walberer and coworkers defined isostericity from a theoretical analysis [22]. They generated a base pair database for all possible hydrogen bond arrangements and deduced an isostericity measure based on glycosidic bond overlap. They observed high isostericity values in helical base pairs, validating their approach, and more surprisingly in several purine–purine and pyrimidine–pyrimidine combinations.

Accurate RNA sequence comparison requires precise (structural) alignments in order to ensure the positions compared truly correspond in the tertiary structure. Including isosteric information in sequence alignment gives better insights into the sequence requirement of structural motifs (see Section 3) across RNA phylogenies [23]. The presence of isosteric base pairs is another fact that supports a structural rather than sequential RNA evolution.

3 Motif Discovery

In the previous section, we presented a series of nomenclature and formalisms to describe some already acknowledged components of RNA tertiary structure: the nucleotide conformations and interactions. Here, we take one step further and describe higher-order RNA components.

The increase in high-resolution X-ray crystallographic structures, in particular the resolution of the large ribosomal subunit [24–26], has increased the literature describing repeated RNA fragments or motifs [27]. Many occurrences of each of these fragments can be found in one or among several different 3-D structures, they are conserved among evolutionary related RNAs, and they are often related to specific structural and biochemical functions.

One can think of RNA motifs as fundamental RNA building blocks. Therefore, finding and characterizing all of them should provide us with invaluable knowledge about RNA folding and aid substantially in tertiary structure prediction. Here, we present some classical RNA motifs, and introduce a formal definition allowing us to computationally represent, search for and discover them.

3.1 RNA Motifs

The most obvious RNA motif is the double helix, which is composed of a succession of stacked Watson–Crick base pairs. Similarly to the double helix, RNA motifs are thermodynamically stable and fold into similar tertiary structures that can be found in various structural contexts.

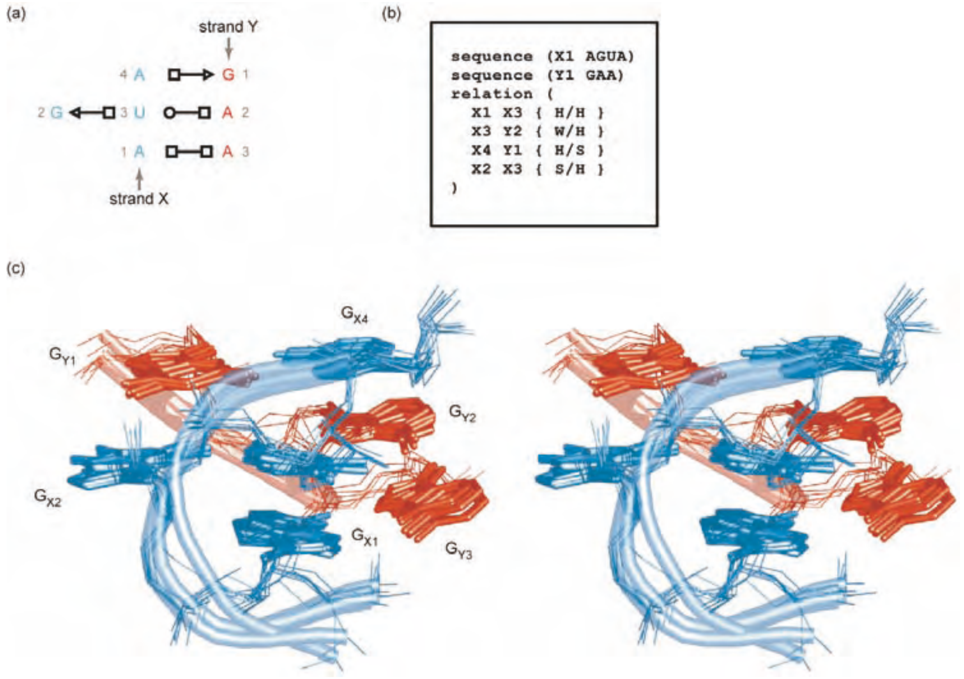
Let us define an RNA motif as a graph of nucleotide conformations and interactions, where the nucleotides are the vertices of the graph. An arc between two nucleotides is present if the two nucleotides are adjacent in the sequence or if their bases interact. Note that if we use the nomenclature introduced in the previous section, then this definition is equivalent to our formal representation of an annotated RNA tertiary structure and is, in fact, the output of the MC-Annotate computer program.

While RNA graphs are easily represented in computer programs by classical data structures, there is currently no consensus in the RNA ontology nor is there a data file format to represent them. RNA graphs in computer programs such as MC-Annotate are serialized into opaque binary files using the C++ MC-Core library developed in our laboratory (also freely available at sourceforge.net/projects/mccore). The RNAML format, derived from XML (extensible markup language), can handle RNA graphs and is portable among many different RNA applications [28,29].

Since RNA motifs can be represented by characteristic RNA graphs, they can be searched within hosting RNA tertiary structures via graph isomorphism; the occurrences of an RNA motif are simply the isomorphic subgraphs in the hosting graphs. Our laboratory implemented the classical graph isomorphism algorithm [30] in a computer program called MC-Search. The input to MC-Search is a description of the target RNA structure, or pattern, from which MC-Search returns all occurrences of the target motif found in a set of pre-selected PDB files.

3.1.1 Classical Examples

Consider the sarcin/ricin motif (Figure 10), which has been predicted to occur in many different locations of the large ribosomal subunit by comparative sequence analysis [31]. The MC-Search descriptor file for the sarcin/ricin motif is shown in Figure 10(b). MC-Search finds seven occurrences of this motif in the crystal structure of the *H. marismortui* 23S rRNA (PDB ID 1JJ2). All occurrences found share a maximum RMSD of 0.93 Å. Interestingly, the annotation of the found occurrences revealed four conserved base stacking interactions (see Figure 10c), including those among nonadjacent nucleotides:



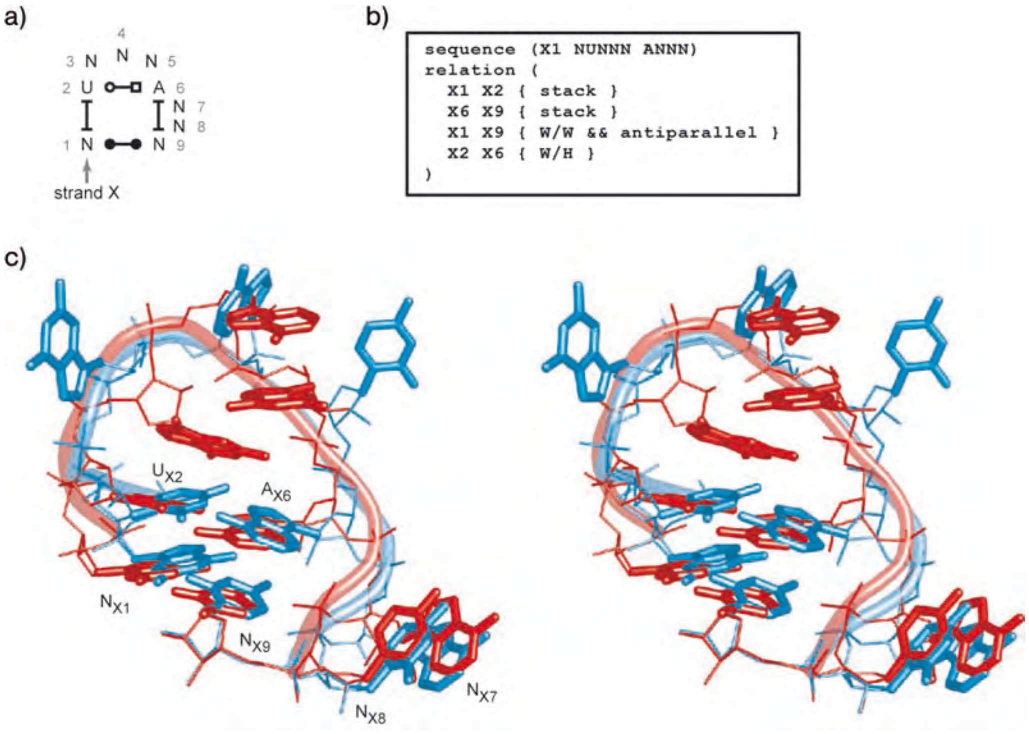
```
sequence (X1 AGUA)
sequence (Y1 GAA)
relation (
  X1 X3 { H/H }
  X3 Y2 { W/H }
  X4 Y1 { H/S }
  X2 X3 { S/H }
)
```

Figure 10 Sarcin/ricin motif. The motif is made of two strands, here named X (shown in blue) and Y (red), and their nucleotides, respectively, numbered 1–4 and 1–3. (a) A schematic representation using *LW* (circle for W; square for H; triangle for S). (b) MC-Search input. (c) The seven occurrences in *H. marismortui* 23S rRNA crystal structure optimally aligned (stereoview). The threads follow the phosphorus atoms in each strand. The hydrogen atoms are not shown.

A_{X1} <> U_{X3} (outward), G_{X2} >< G_{Y1} (inward) and A_{X4} <> A_{Y2} (outward). Here, the two strands involved in the motif were arbitrarily named X and Y, and their nucleotides numbered, respectively, 1–4 and 1–3.

A motif corresponding to the T-loop conserved across tRNAs was matched in the ribosome. In tRNAs, the loop capping the T-stem is characterized by a *trans* W/H U–A base pair stacked on the last W/W G=C base pair of the stem with a two- or three-nucleotide bulge on the A side. Several instances of the motif where found by visual inspection in *H. marismortui* 23S and in *Thermus thermophilus* 16S subunits [32]. These tRNA T-loop motifs in the ribosome were found to interact with other elements of their rRNA through tertiary interactions, similarly to the interactions found between the T- and the D-loop in tRNAs. Two instances of the two-nucleotide bulge version are found by MC-Search in the *H. marismortui* 23S subunit (Figure 11).

The frequently observed A-minor motif [33] is made of an adenine that interacts with the minor groove of a double helix and is of particular interest since it is involved in the selection of tRNAs by the ribosome [34]. Nine

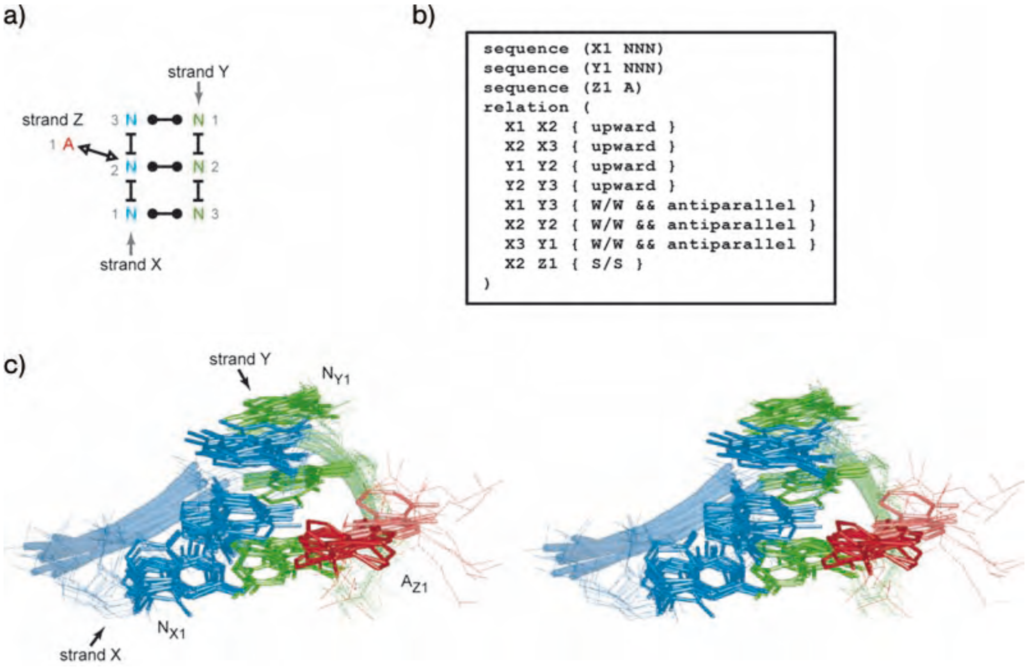


```
sequence (X1 NUNNN ANNN)
relation (
  X1 X2 { stack }
  X6 X9 { stack }
  X1 X9 { W/W && antiparallel }
  X2 X6 { W/H }
)
```

Figure 11 T-loop motif. The motif is made of one strand, here named X, and the nucleotides, respectively, numbered 1–9. (a) A schematic representation using *LW* (circle for *W*; square for *H*; triangle for *S*). (b) MC-Search input. (c) The two occurrences in *H. marismortui* 23S rRNA crystal structure (PDB 1JJ2) optimally aligned (stereoview). The instance found at X1 = 334 is shown in blue; the one at X1 = 1387 in red. The threads follow the phosphorus atoms in each strand. The hydrogen atoms are not shown.

instances of the A-minor motif are found by MC-Search in the *H. marismortui* 23S subunit (Figure 12). Note that the *S* edge in the base pair annotation of the double-helix nucleotide indicates the minor groove interaction. The A-minor motif is a key element of the larger K-turn motif which induces a bend between two double helices [35]. The core of the K-turn motif is constituted of two *S*/*H* G–A base pairs. There is only one occurrence of the K-turn motif in the *H. marismortui* 23S subunit (Figure 13).

The tetraloop/receptor motif is most frequent in RNAs. It stabilizes the conformation of a hairpin loop interacting with the minor groove of a stem. It was discovered in the hammerhead ribozyme [36] and in the group I intron [37]. The tetraloop/receptor participates in protein translation fidelity, and in the association of the rRNA 16S and 30S subunits, as mutations in the motif induce loss of ribosomal activity [38]. In *T. thermophilus* rRNA, a conserved

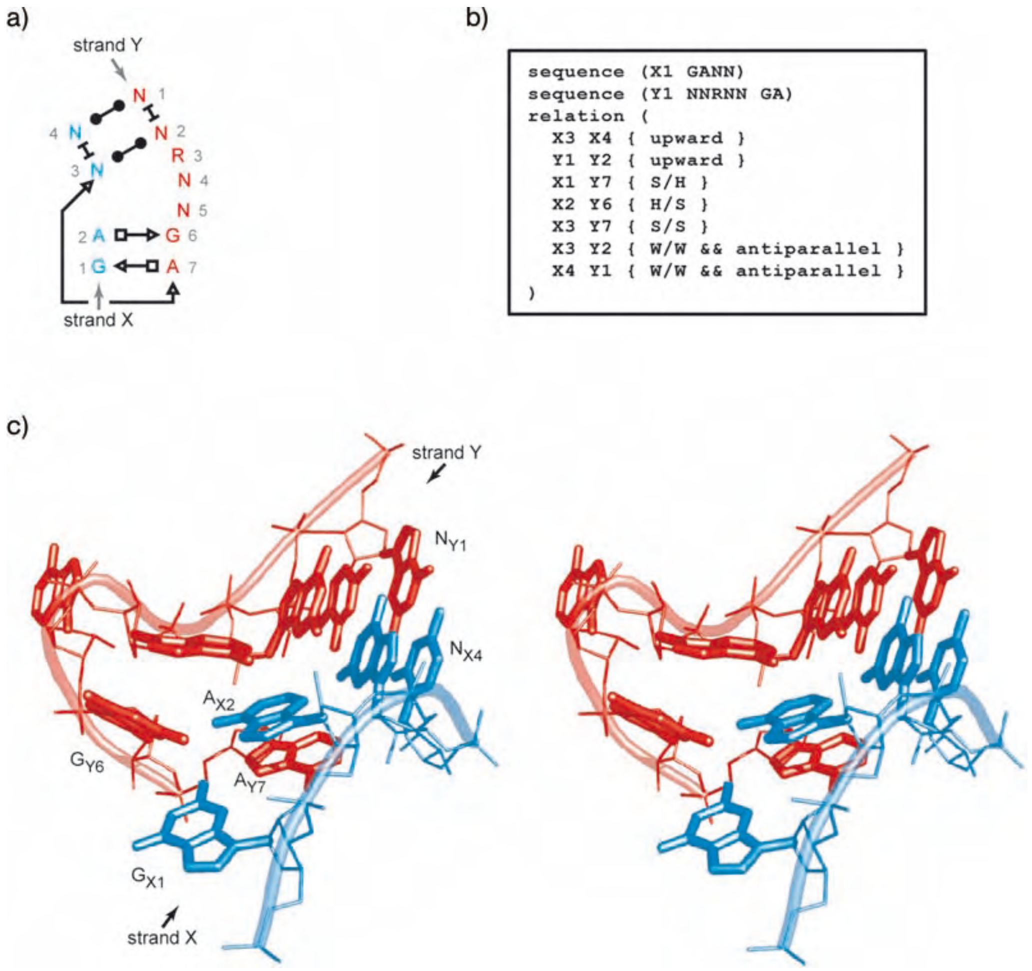


```
sequence (X1 NNN)
sequence (Y1 NNN)
sequence (Z1 A)
relation (
  X1 X2 { upward }
  X2 X3 { upward }
  Y1 Y2 { upward }
  Y2 Y3 { upward }
  X1 Y3 { W/W && antiparallel }
  X2 Y2 { W/W && antiparallel }
  X3 Y1 { W/W && antiparallel }
  X2 Z1 { S/S }
)
```

Figure 12 A-minor motif. The motif is made of three strands, here named X (shown in blue), Y (green) and Z (red); the nucleotides are, respectively, numbered 1–3, 1–3 and 1. (a) A schematic representation using *LW* (circle for W; square for H; triangle for S). (b) MC-Search input. (c) The nine occurrences in *H. marismortui* 23S rRNA crystal structure optimally aligned (stereoview). The threads follow the phosphorus atoms in each strand. The hydrogen atoms are not shown.

GCAA tetraloop (referred to as loop 900) caps helix 27 in the 16S subunit and binds to the minor groove of helix 24 in the 30S subunit. Two occurrences of the tetraloop/receptor motif-binding stems of at least three W/W base pairs are found by MC-Search in the *H. marismortui* 23S subunit (Figure 14). Interestingly, the two occurrences found have different base pair orientations.

As found in many pathogenic viruses, the formation of a pseudoknot motif in mRNAs may induce frameshifting in the protein translation [39,40]. The pseudoknot is made of a hairpin stem–loop whose nucleotides in the loop participate in the formation of a second stem. In fact, a pseudoknot occurs when the four strands, A5′, A3′, B5′ and B3′, involved in the formation of two stems, A and B, interleave in the sequence: A5′–B5′–A3′–B3′. Pseudoknots are reported by the MC-Annotate computer program, and 15 are found in the crystal structure of the *H. marismortui* 23S rRNA (PDB ID 1JJ2). In feline immunodeficiency virus (FIV), a pseudoknot was found by comparative sequence and mutagenesis analyses [41]. The secondary and tertiary structures of the FIV pseudoknot are shown in Figure 15. See Section 4.3.2 for details



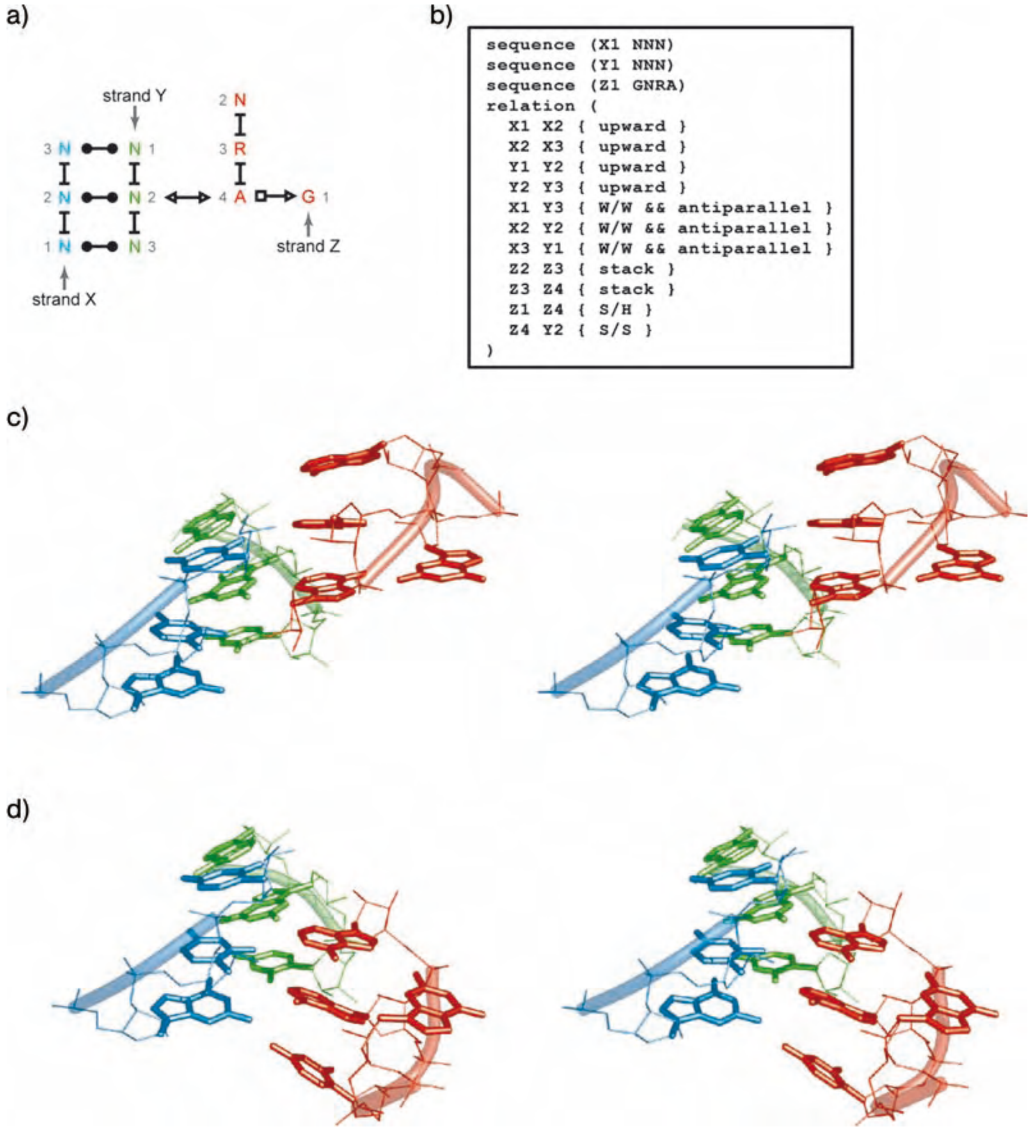
```
sequence (X1 GANN)
sequence (Y1 NNRNN GA)
relation (
  X3 X4 { upward }
  Y1 Y2 { upward }
  X1 Y7 { S/H }
  X2 Y6 { H/S }
  X3 Y7 { S/S }
  X3 Y2 { W/W && antiparallel }
  X4 Y1 { W/W && antiparallel }
)
```

Figure 13 K-turn motif. The motif is made of two strands, here named X (shown in blue) and Y (red), and their nucleotides, respectively, numbered 1–4 and 1–7. (a) A schematic representation using *LW* (circle for W; square for H; triangle for S). (b) MC-Search input. (c) The one occurrence in *H. marismortui* 23S rRNA crystal structure (stereoview). The threads follow the phosphorus atoms in each strand. The hydrogen atoms are not shown.

how Fabris and his collaborators combined mass spectroscopy and computer modeling to determine the FIV pseudoknot tertiary structure [42].

3.2 Catalytic Motifs

The structure and function of catalytic RNAs (ribozymes) have extensively been studied [43–51]. In particular, the crystal structure of the hammerhead ribozyme shows a three-way junction catalytic core [45]. The three-way junc-



```
sequence (X1 NNN)
sequence (Y1 NNN)
sequence (Z1 GNRA)
relation (
  X1 X2 { upward }
  X2 X3 { upward }
  Y1 Y2 { upward }
  Y2 Y3 { upward }
  X1 Y3 { W/W && antiparallel }
  X2 Y2 { W/W && antiparallel }
  X3 Y1 { W/W && antiparallel }
  Z2 Z3 { stack }
  Z3 Z4 { stack }
  Z1 Z4 { S/H }
  Z4 Y2 { S/S }
)
```

Figure 14 Tetraloop receptor. The motif is made of three strands, here named X (shown in blue), Y (green), and Z (red); the nucleotides, respectively, numbered 1–3, 1–3 and 1–4. (a) A schematic representation using *LW* (circle for *W*; square for *H*; triangle for *S*). (b) MC-Search input. (c) The occurrence in *H. marismortui* 23S rRNA crystal structure at position X1 = 1552, Y1 = 1567 and Z1 = 1629. The Z4–Y2 base pair is parallel. (d) The occurrence in *H. marismortui* 23S rRNA crystal structure at position X1 = 2529, Y1 = 2490 and Z1 = 1055. The Z4–Y2 base pair is antiparallel. The threads follow the phosphorus atoms in each strand. The hydrogen atoms are not shown.

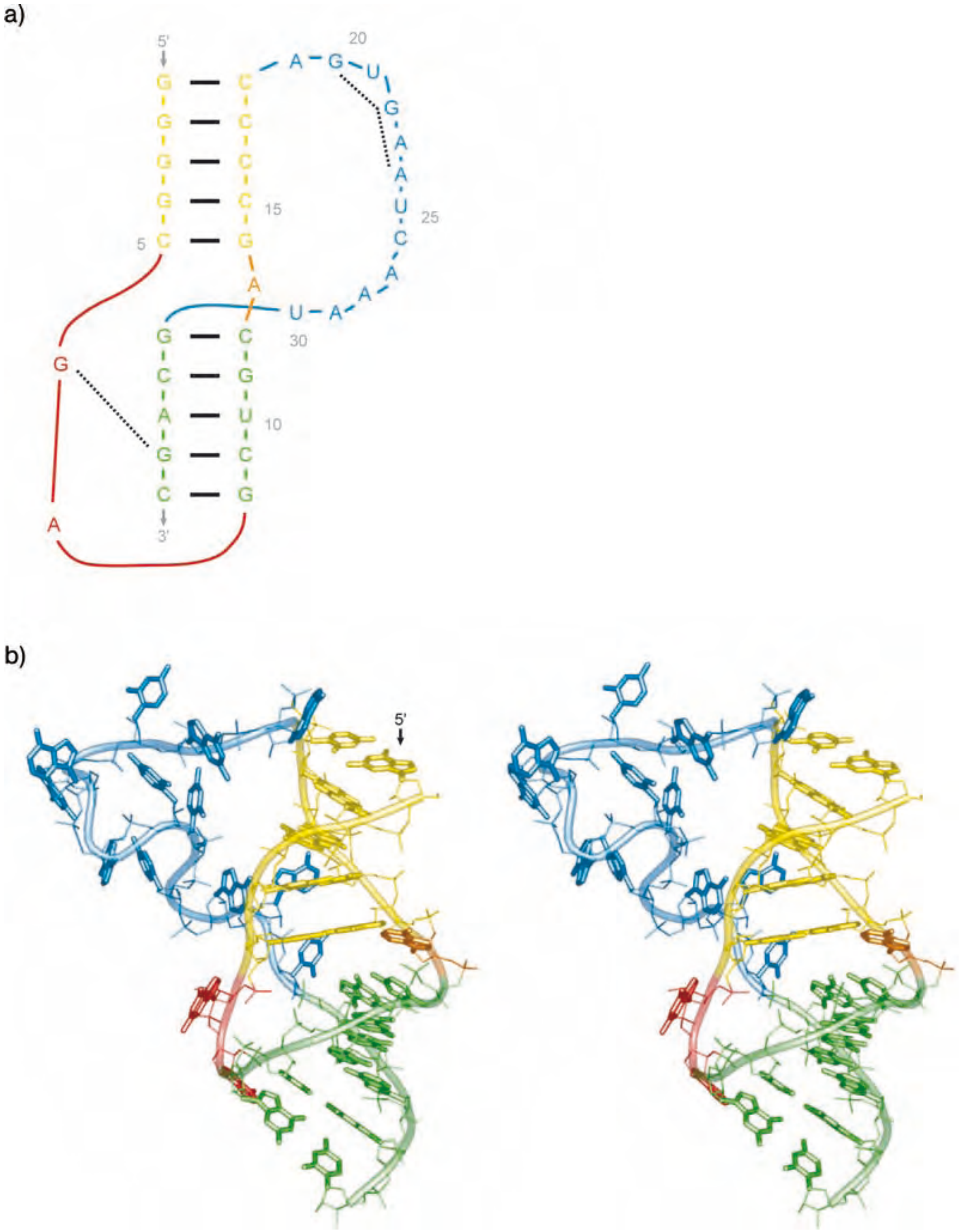


Figure 15 FIV pseudoknot structure. (a) Secondary structure. The phosphodiester linkage is shown by thin lines. The W/W base pairs are shown by thick lines. The tertiary interactions predicted by chemical cross-linking are shown in dashed lines. (b) Tertiary structure (stereoview). The thread shows the phosphodiester backbone. The hydrogen atoms are not shown.

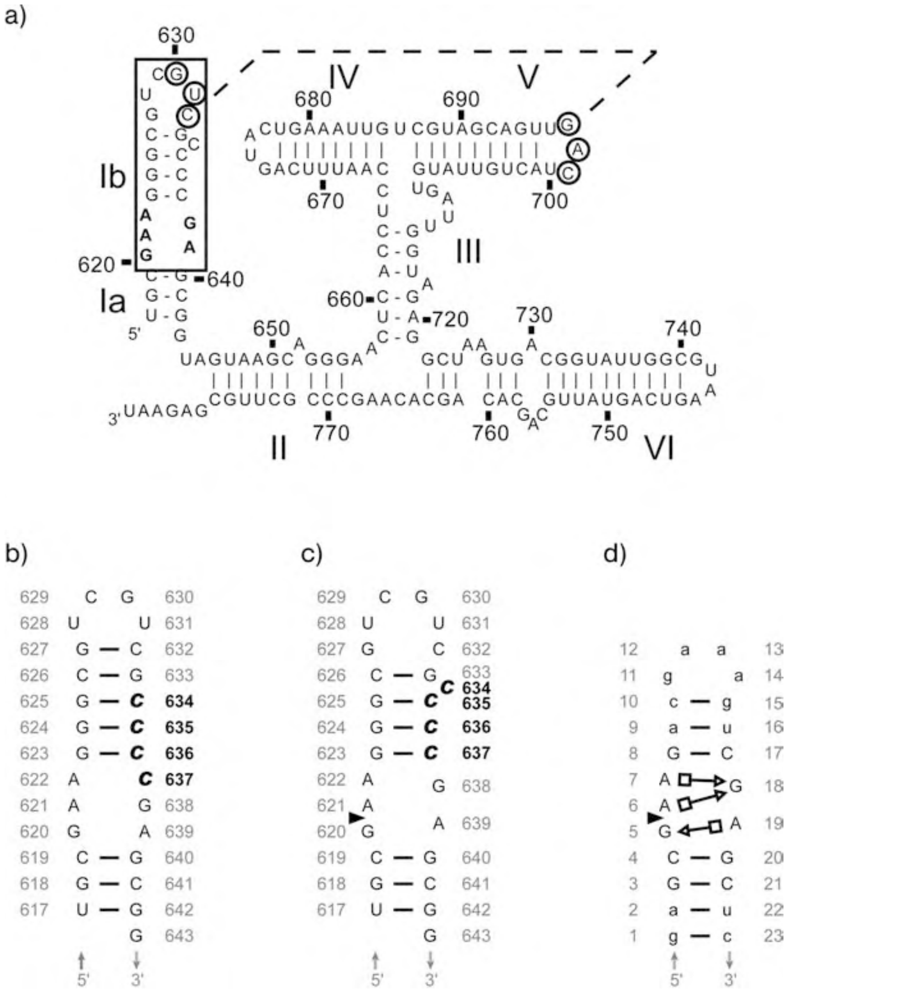
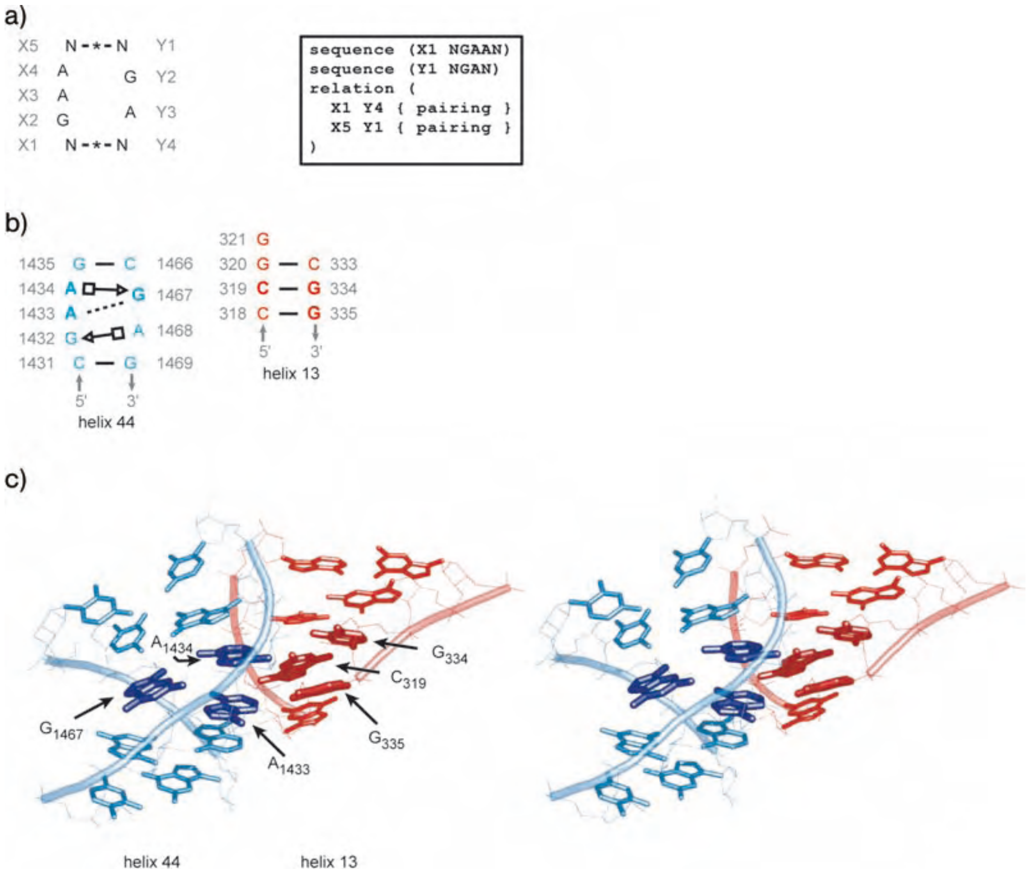


Figure 16 *Neurospora* VS ribozyme. (a) Secondary structure. The substrate domain is shown in the box. The dotted line indicates the tertiary interactions of the circled bases. (b) Inactive state of the substrate domain. C_{637} becomes the base pair partner of G_{623}, which affects the base pair registry of stem I. C_{637} to C_{634} (shown in bold) are shifted by one base pair towards the 5′ strand. As a result, C_{634} looses its base pair partner in the active state. (c) Active state. The arrow points the cleavage site. (d) Mutant stem–loop I (SL1′). Uppercase nucleotides belong to the wild-type; lowercase to mutations. The sheared G–A base pairs identified in the NMR structure are shown using the *LW* nomenclature (circle for *W*; square for *H*; triangle for *S*).

tion motif is made of three double helices, and its topology has been much studied by Lescoute and Westhof [52]. Interestingly, the H/H and S/S loop–loop interactions involved in the catalytic mechanism of the hammerhead have been characterized by 3-D modeling by Massire and Westhof, using their computer program MANIP [53,54].



```
sequence (X1 NGAAN)
sequence (Y1 NGAN)
relation (
  X1 Y4 { pairing }
  X5 Y1 { pairing }
)
```

Figure 17 VS ribozyme SL1′ active internal loop motif. (a) MC-Search input. The strands are named ”X” and ”Y”, respectively, for the 5′ and 3′ strands. A star in the schema (left) is used as a wildcard matching any type of base pairs, which is denoted “pairing” in the input file (right). (b) The secondary structure of the occurrence found in *T. thermophilus* 16S rRNA. *LW* is used (circle for *W*; square for *H*; triangle for *S*). This motif matches in helix 44 of the 16S rRNA (left) and forms many ribose–ribose contacts and two A-minor motifs with helix 13 (right). (c) Stereoview of helix 44 (blue) and helix 13 (red). The match is at 1.40 Å of RMSD of the NMR structure of the VS ribozyme SL1′ loop. The shared sheared G–A base pairs are shown in bold. The nucleotides in helix 13 that participate to the A-minor motifs are shown in bold.

Another relationship between the structure and catalytic activity of RNAs was discovered in the *Neurospora* Varkud Satellite (VS) ribozyme. Six helical domains characterize the secondary structure of the self-cleaving VS ribozyme (see Figure 16a). The substrate domain at stem–loop I is recognized by the catalytic core by, so far, unknown loop–loop interactions between stem–loop I and V [55]. The cleavage mechanism of this ribozyme is induced by a modification of the base pair registry in stem I (see Figure 16b and c). C_{637}

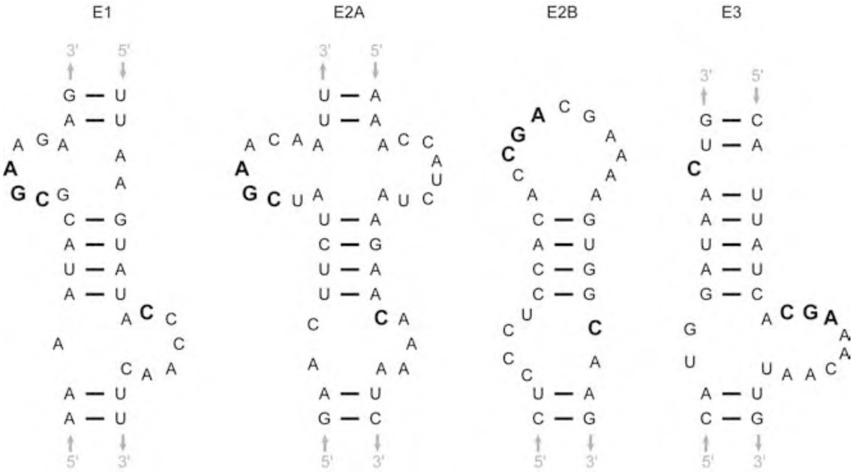


Figure 18 Secondary structures of the four localization elements within *ASH1* mRNA in yeast *S. cerevisiae*: E1, E2A, E2B and E3. Only fragments hosting the loop–stem–loop common motif are shown, with the conserved CGA and C shown in bold.

is paired to G_{623} in the active state, reducing the size of the 3′ strand of the internal loop from three to two.

To get a deeper insight into the 3-D structure of the VS ribozyme active conformation, a nuclear magnetic resonance (NMR) spectroscopy structure of a stem–loop I mutant (SL1′) was built to mimic the active conformation [56]. In previous NMR structures of the inactive stem–loop I, the 3–3 internal loop was composed of three stacked base pairs, two S/H G–A base pairs and a wobble A–C base pair. In the SL1′ active structure, the A–C base pair is broken by the helix shift, resulting in a double S/H base pair shared between both As on the 5′ strand and the G on 3′ strand (see Figure 16c).

This distinctive internal loop motif was searched in other RNAs with the MC-Search computer program without any specific interaction types in the loop, by using the "pairing" keyword. The pattern matched a fragment in helix 44 of the *T. thermophilus* 16S rRNA, which is at 1.4 Å of RMSD to the NMR structure of the interior loop of SL1′ (see Figure 17). Helix 44 in the rRNA interacts with helix 13 to form a ribose zipper motif [57], which is defined by ribose–ribose interactions and the presence of the A-minor motif. Both adenines share one sheared (S/H) G–A base pair. The SL1′ structure became a model for the active substrate element of the VS ribozyme, which can bind to another helix and form a ribose zipper motif. Since the catalytic mechanism is characterized by tertiary interactions between stem–loop I and V (see Figure 16a), the authors suggest that the specific interaction needed for an accurate recognition of the substrate is this ribose zipper.

3.3 Transport and Localization

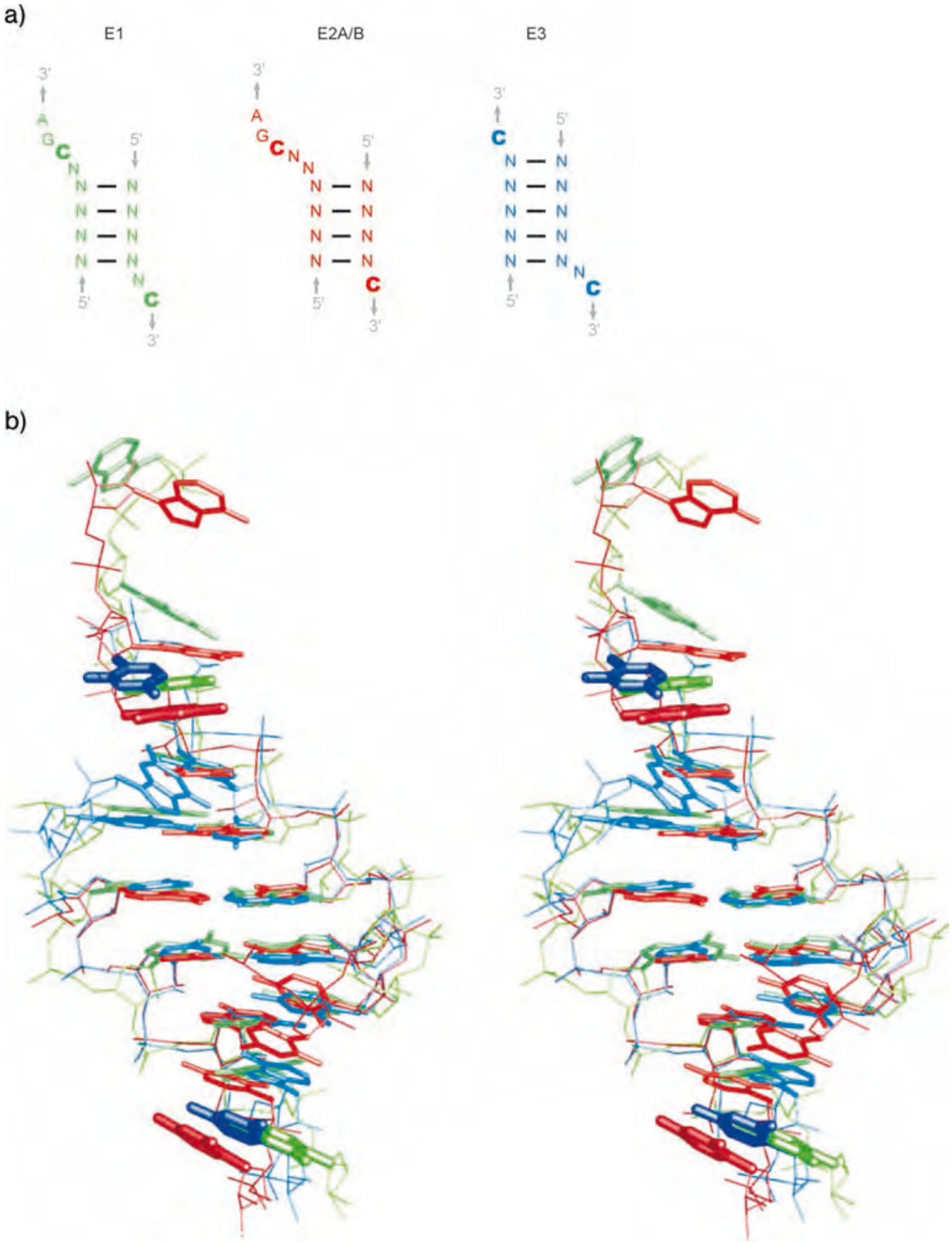
A joint effort between Chartrand's laboratory in Montreal and our group resulted in a deeper understanding of the molecular basis behind cytoplasmic mRNA transport and localization to the yeast bud [58]. A small RNA motif was found conserved across four localization elements within the mRNA of the *ASH1* protein. The motif was found to interact with She2p, one of the important components of the yeast mRNA localization machinery in *Saccharomyces cerevisiae*.

Three localization elements occur in the coding region (E1, E2A and E2B) [59] and the fourth is located in the 3′ untranslated region (E3) [60]. Conserved nucleotides in each element were identified by *in vivo* selection from a polymerase chain reaction (PCR) library. Similar She2p RNA motif-binding sites were characterized, and their secondary structures predicted. The generalized motif is composed of two loops, separated by a short stem, which contains a conserved C on the 3′ strand and a CGA sequence on the 5′ strand (see Figure 18), but for E3 for which the loops are inversed.

A structural rule of the generalized She2p-binding motif was deduced: *i*:*s*:*j*, where *i* is the number of nucleotides in the loop before the conserved CGA sequence (*cf.* one in the 5′ loop of E1: GCGAAGA), *j* is the number of nucleotides before the conserved C in the 3′ loop (*cf.* one in the 3′ loop of E1: ACCCAAC) and *s* the length of the stem separating the two loops (*cf.* four in E1). The localization element E1 is thus a 1:4:1 motif, E2A and E2B are 2:4:0, and E3 is a 0:5:1. The sum $i + s + j = 6$ is invariant.

To find a structural explanation of the invariant rule, MC-Search was used to scan the high-resolution RNA 3-D structures to seek occurrences of these three types of She2p-binding motifs (see Figure 19). In total, 123 matches were found for type 1:4:1 (E1), 85 for type 2:4:0 (E2A/E2B) and 717 for type 0:5:1 (E3). The distance between the 3′ phosphate groups of the two conserved cytosines in all occurrences was 28.3 ± 0.9 Å for type 1:4:1, 28.0 ± 1.0 Å for type 2:4:0 and 28.2 ± 0.7 Å for type 0:5:1. Increasing the length of the She2p-binding motif stem in each localization elements in the yeast three-hybrid assay resulted in a total loss of interaction with She2p.

The resolution of the X-ray crystal structure of She2p revealed a helical region essential for its interaction with *ASH1* mRNA localization elements. This helical region covers a distance of about 27 Å. It was thus logical to the authors to propose a model of the *ASH1* mRNA localization element motif binding to this region of the protein through interactions between the two conserved Cs. An interesting conclusion from this project is that tertiary structure conservation of the RNA motif is more relevant to the binding function than sequence conservation.


a)
E1
E2A/B
E3
3'
A
G
C
N
N — N
N — N
N — N
N — N
N
C
5'
3'
b)

4 Modeling

In the previous sections, we introduced an ontology to model RNA tertiary structure components and motifs. Here, we employ the term "modeling" to describe 3-D model building. Two types of 3-D models exist: physical or handicraft models (wood, plastic, metal or any other artistic materials) and computer models (interactively built or directly generated). Often, scientists start with the former type, mainly to draft ideas and to get a global picture of the RNA tertiary structure, and then, when enough structural data are gathered, they switch to the latter.

The goal of RNA tertiary structure modeling is to summarize and project in 3-D structural data. In the last few decades, many low-resolution techniques, e.g. footprinting, and enzymatic and chemical probing, have improved and produced a large quantity of structural data. Consequently, RNA tertiary structure modeling has become very popular and useful in recent years to translate these structural data in to precise 3-D models. The low-resolution techniques compensate when higher-resolution ones cannot be applied, e.g. in the case of *in vivo* or reactive conformations.

In this section, we focus on computer-assisted model building. There are two major approaches to 3-D modeling. The first approach starts with all nucleotides in an extended or randomized state, which is then successively modified until a folded and satisfactory state is reached. The conformational space of such folding methods is defined by the number of accessible states, e.g. molecular mechanics is a folding method that defines satisfaction as the optimum of an objective function. Harvey's laboratory has developed an RNA objective function composed of penalty terms corresponding to experimentally determined nucleotide interactions and distances. The objective function is penalized if the observed interactions and distances are not present in a state. They simplified the RNA model by using one to five points per nucleotide to speedup the folding operations. Using YAMMP, their computer program, they were able to build models of large ribosomal subunits [61]. In the second approach, one assembles the components of the RNA by using

←

Figure 19 The She2p-binding motif. (a) MC-Search input for the three motif types, from left to right 1:4:1 (E1), 2:4:0 (E2A/B) and 0:5:1 (E3). The conserved Cs are shown in bold. (b) Stereoview of three occurrences, one for each type, aligned by their stems. Type E2A/B is shown in red and was found in *H. marismortui* 23S rRNA (PDB ID 1M1K, strands 1463–1467 and 1477–1485 in chain 'A'). Type E3 is shown in blue and was found in *Deinococcus radiodurans* 23S rRNA (PDB ID 1JZY, strands 295–300 and 363–369 in chain "A"). Finally, type E1 is shown in green and was found in a NMR structure of a hairpin similar to the P5abc region within group I intron (PDB ID 1EOR, strands 4–11 and 16–21 in chain "A"). The conserved Cs are shown in bold in the three models.

construction operators to position and orient them in 3-D space. MANIP [54], an interactive system, and MC-Sym [62], an automated procedure, are among the most employed computer systems to model RNA tertiary structures.

In this section, we present how RNA tertiary structure modeling can be mapped to the discrete constraint satisfaction problem (CSP) and how the CSP solver was implemented in the MC-Sym computer program.

4.1 The CSP

The CSP can be described by three finite sets: the variables $V = \{v_1, v_2, \dots, v_n\}$, the domains $D = \{d_1, d_2, \dots, d_n\}$ and the constraints $C = \{c_1, c_2, \dots, c_m\}$ [63,64]. A variable v_i is assigned values from domain $d_i = \{d_{i,1}, d_{i,2}, \dots, d_{i,|d^{i}|}\}$. Each constraint restricts the assignment of a subset of V, called the constraint scope. For example, if the scope of $c_1 \in C$ is $\{v_2, v_4, v_5\} \subset V$, then $c_1 \subset d_2 \times d_4 \times d_5$.

The three sets are defined in the context of the problem application. In RNA tertiary structure prediction, the variables could be the nucleotides and the domains their possible 3-D coordinates. An example of constraint would be two nucleotides that form a base pair. The scope of the base pair constraint is the two nucleotide partners which link their relative positions and orientations in 3-D space.

A solution to the CSP is a complete variable assignment, $v_i \in d_i$ for $i \in \{1 \dots n\}$, so that all constraints in C, $c_j \in C$ for $j \in \{1 \dots m\}$ are satisfied. A constraint c_j is satisfied only if its scope in the solution is assigned according to the relation it defines. Solving the CSP consists in finding one, many or all solutions.

The search space of a CSP is the Cartesian product of all d_i:

$$\prod_{i=1}^{n} |d_i| \, . \tag{4}$$

The size of a CSP search space is exponential in the number of variables domain sizes. The solutions of a CSP are found by exploring the variable assignments of its search space and verifying if they satisfy the constraints. Backtracking is the classical search algorithm to solve a CSP deterministically and exhaustively. In backtracking, the variables are assigned values systematically, one at a time and to the next available value from its associated domain. When all the values of a domain have been tried, the domain is reset and backtracking moves to the previous variable, assigning its next value, before continuing. The search finishes when the domain of the first variable is reset, indicating that all possible assignments have been tried.

Backtracking develops a search tree (see Figure 20), where the nodes are visited in a depth-first manner. A path from the root of the search tree to a leaf represents a solution if all of its nodes are consistent (the black nodes in

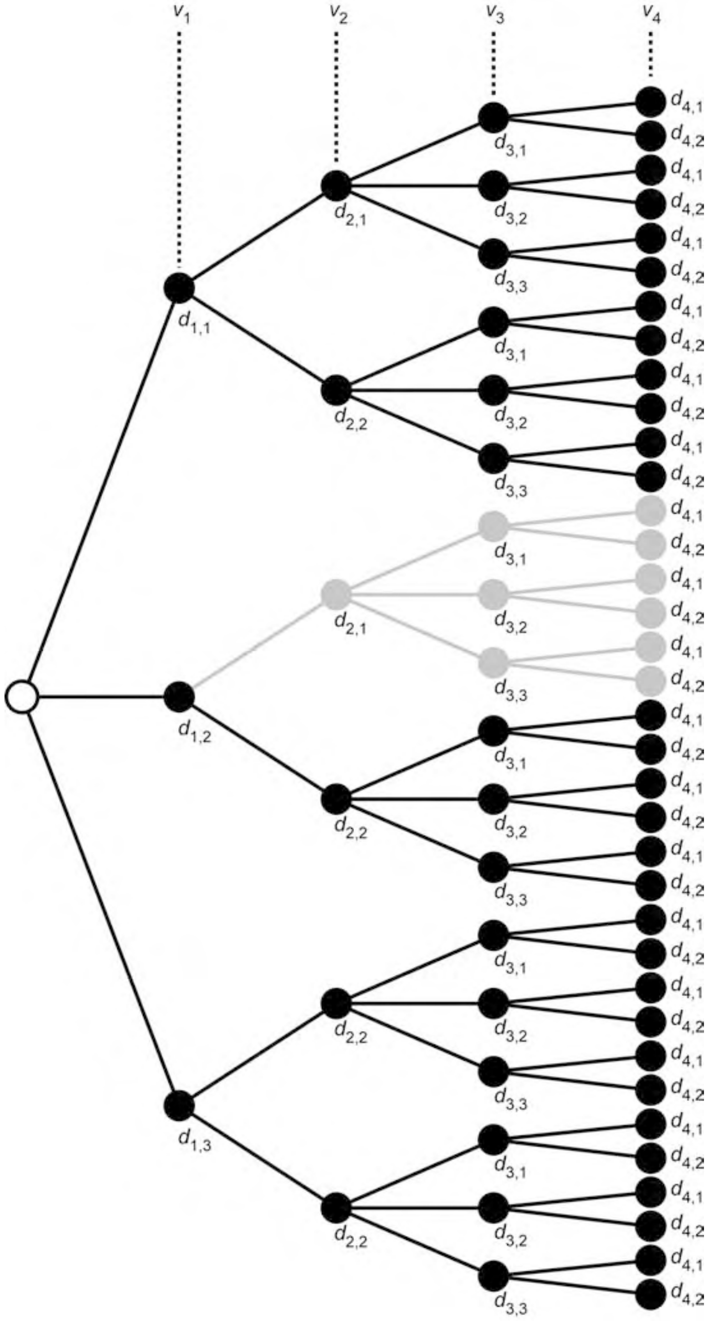


Figure 20 Backtracking search tree. The search space is defined by $V = \{v_1, v_2, v_3, v_4\}$ and $D = \{d_1, d_2, d_3, d_4\}$, where $d_1 = \{d_{1,1}, d_{1,2}, d_{1,3}\}$, $d_2 = \{d_{2,1}, d_{2,2}\}$, $d_3 = \{d_{3,1}, d_{3,2}, d_{3,3}\}$ and $d_4 = \{d_{4,1}, d_{4,2}\}$. Any path from the root (empty circle) to a leaf assigns each variable to a value from its respective domain. A verification of the constraint $\{(v_1, v_2) \in d_1 \times d_2 \quad | \; (v_1, v_2) \neq (d_{1,2}, d_{2,1})\}$ prunes the subtree in grey, as soon as the backtracking assigns v_2 to $d_{2,1}$, and then the search jumps to the next assignment for v_2.

Figure 20). The search tree size is given by Eq. (5):

$$1 + \sum_{i=1}^{n} \sum_{j=1}^{i} |d_i| \ . \tag{5}$$

However, the search can avoid visiting most inconsistent nodes by pruning from the search tree inconsistent branches (those with grey nodes in Figure 20), as soon as one inconsistent node is found. This trick does not change the search time complexity of backtracking, which is exponential in n and $|d_i|$, but can reduce the search time in practice. Using MC-Sym (see next section), the search space of a particular instance of the CSP for the yeast tRNA-Phe tertiary structure was 10^{26}, but the constraints used to explore it defined a consistent search tree of only about 5×10^5, which was explored in seconds and included only about 30 solutions [65]. The best model was at approximately 3.0 Å of RMSD to the yeast tRNA-Phe, as well as with the yeast tRNA-Asp crystal structures (crystal structure resolution). In RNA tertiary structure prediction, it is customary to assess the quality of a prediction method by evaluating its performances at refolding known structures, and RMSD is a very popular quantitative measure.

4.2 MC-Sym

MC-Sym (Macromolecular Conformations by SYMbolic programming) is a computer program for building RNA tertiary structure models from syntactic descriptions of the RNA, similar to the ones used for MC-Search. The computer program implements a CSP solver with backtracking and thus generates models that are consistent with the constraints [6,62,65,66].

In MC-Sym, the variables, V, correspond to the vertices, v, and arcs, a, of the RNA graph (see Section 3.1). The domains of the vertices, D_v, are rigid nucleotides (the relative 3-D atomic coordinates never change) and those of the arcs, D_a, are linear transformation matrices encoding base stacking and base pairing. Consequently, the domains, D, are taken from the Cartesian product, $D = D_v \times D_a$. The goal is to generate consistent 3-D atomic coordinates to each nucleotide in the global frame of the RNA. The rigid sets of nucleotide 3-D atomic coordinates, as well as the linear transformation matrices, are extracted from the PDB [1]. The nucleotide is defined by cutting the phosphodiester chain at the O3′–P bond.

The linear transformation matrices combine nucleotide rotations and translations in 4-D homogeneous coordinates, which represent the spatial relation between two stacked or paired bases. As we saw in Section 2.2, nucleotide interactions involve their bases and thus we represent their spatial relations by coordinate frame relative transformations. The frames are defined at the

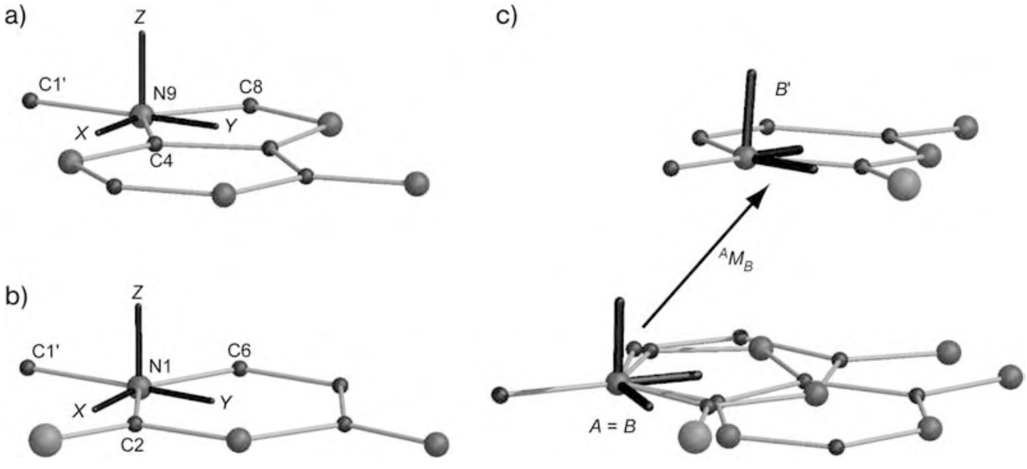


Figure 21 Nucleotide coordinate frame. Hydrogen and backbone atoms are not shown. (a) Frame for a purine (here an adenine). The origin is on the N9 atom, and the *XY* plane is aligned on the plane described by atoms N9, C4 and C8, shared by all purines. (b) Frame for a pyrimidine (here a cytosine). The origin is on the N1 atom, and the *XY* plane is aligned on the plane described by atoms N1, C2 and C6, shared by all pyrimidines. (c) Transformation of cytosine B in adenine A's frame that expresses a stacking interaction. B is moved from a position aligned on A's local frame to B' by applying $^{A}M_{B}$.

terminal nitrogen atom (N9 in purines and N1 in pyrimidines), with the XY plane aligned with the base plane (see Figure 21a and b).

For any coordinate frames A and B, the relative transformation $^{A}M_{B}$ expresses B's position (right subscript) in A's coordinates (left superscript). For example, let $^{A}M_{B}$ be a relative transformation that expresses a stacking interaction between an adenine (frame A) and a cytosine (frame B). As the relative transformation $^{A}M_{B}$ is defined in A's frame, B can be moved from a position aligned on A's frame to a position that expresses the stacking interaction between the two bases (Figure 21c). This relative transformation places B in the local frame of A, independently of A's position in the global frame. Then, to obtain B's position in the global frame, O, it must be transformed by the relative matrix $^{O}M_{A}$, which expresses A's position in the global frame.

Using such transformations, all nucleotides of an RNA graph can be positioned relative to another by starting with an initial origin nucleotide, whose frame defines the global frame, O. A construction order is defined by selecting a spanning tree of the RNA graph rooted at the origin nucleotide. The transformations of the arcs of the spanning tree applied to the rigid coordinates of the vertices build the RNA (see Figure 22). Ideally, the chosen spanning tree of the RNA graph would include all arcs. However, RNA graphs are rarely

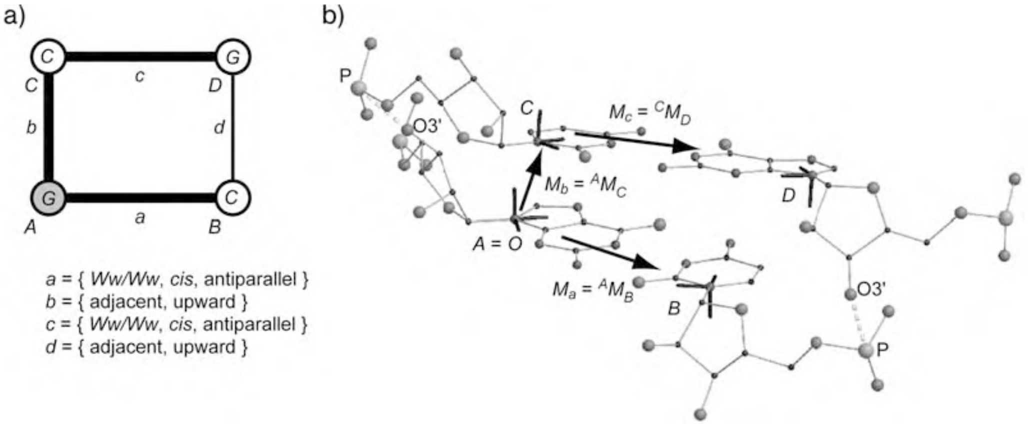


Figure 22 A two-stacked G=C base pairing CSP. (a) RNA graph. The vertices are in uppercase; arcs in lowercase. The arcs of a chosen spanning tree are shown in bold; rooted at vertex A (shaded). (b) One solution: the nucleotide structure for each vertex and a local transformation matrix for each arc (M_i for arc "i"). The root is aligned to the global coordinate frame O. B is positioned by applying $M_a = {}^AM_B$ in the local frame A, which is aligned with O, and thus B is directly placed in the global frame by M_a; vertex C directly placed by $M_b = {}^AM_C$. Finally, to position D, $M_c = {}^CM_D$ is applied in the local frame of C, which is then moved to the global frame by applying $M_b = {}^AM_C$. The O3′–P bond lengths shown in dashed lines are verified by a distance constraint. In this spanning tree, arc d is not considered in the construction.

trees, [1]as they include cycles of interactions. (The only situation where an RNA graph is a tree is when the only information available is the sequence.) Consequently, the missing arcs must be represented by constraints in the CSP, C. As we will see in Section 4.3.3, this problem can be overcome.

The search space for an RNA tertiary structure prediction problem defined with this instance of the CSP is given by the spanning tree n nucleotide variables and $n - 1$ interactions. Equation (6) gives the size of the search space, the Cartesian product $D_v \times D_a$, where $x_i \in D_v, y_i \in D_a$:

$$\prod_{i=1}^{n} |x_i| \times \prod_{i=1}^{n-1} |y_i| \,. \tag{6}$$

Two types of constraints need to be verified at each variable assignment: the atomic clashes and the O3′–P covalent bond distances. The scope of the atomic clash constraints is all nucleotide pairs and they are needed to insure that no pair of atoms from both nucleotides are overlapping. A threshold inter-atomic distance, typically 1 Å, implements the steric clash constraints.

The adjacency constraint, as we name it, has a scope of two adjacent nucleotides in the sequence and is used to verify the covalent O3′–P bond

[1] The only situation where a RNA graph is a tree is when the only information available is the sequence.

distance (dashed lines in Figure 22b), as the nucleotide backbone are positioned independently. The distance threshold is fixed by a user distance range, typically set to [1,2] Å, as the O–P theoretical distance is approximately 1.6 Å (see Section 4.3). Note that choosing a lower bound smaller that the atomic clash threshold would be useless. Increasing the upper bound is equivalent to "relaxing" the CSP. If an overstretched backbone results from choosing a high upper bound, it can easily be fixed by applying numerical refinement [9].

4.2.1 Backbone Optimization

In the above mapping of the RNA tertiary structure CSP, the adjacency constraint is violated more than 50% of the time, even if its range is relaxed to [1,5] Å. This problem comes from the rigid nucleotide conformations that do not reflect well the overall flexibility of the backbone.

To address this problem, we propose a slight modification of the above CSP mapping. We make the assumption that the RNA tertiary structure is driven by rigid base interactions. Consequently, the backbone conformation can be derived from the bases, and we can define a new CSP on the bases alone. The new CSP defines a search space over the transformation matrices, D_a, reducing considerably its size to $\prod_{i=1}^{n-1} |y_i|$ (see Eq. 6).

To reduce the complexity of building a complete backbone from six free torsion angles (11 in total minus the five of the furanose), we define rigid conformations made of the base and the phosphate group. The ribose interconnects a base with two phosphate groups. To position all of the ribose atoms, six free torsions are needed: θ_0, θ_1, θ_2, θ_3, θ_4 and χ (see Figure 3a). However, we know from Eqs. (1) and (3) that θ_0 to θ_4 are all related by the single pseudorotation angle ρ (Section 2.1). Hence, by fixing all covalent bond lengths and angles to their theoretical values, a ribose structure can be built by optimizing a suitable function, $f(\rho, \chi)$. f builds a ribose using torsions ρ and χ and returns the RMSD between the theoretical and the implicit values of the C5′–O5′ and C3′–O3′ bonds (see Figure 23). Equation (7) gives the value of $f(\rho, \chi)$, where l_k are the measured distances and λ_k are the theoretical distances, $k = 5'$ or $k = 3'$, respectively, for the C5′–O5′ or the C3′–O3′ bond

$$f(\rho,\chi) = \sqrt{\frac{(l_{5'} - \lambda_{5'})^2 + (l_{3'} - \lambda_{3'})^2}{2}}\,. \tag{7}$$

Finding the optimal parameters that minimize Eq. (7) builds a ribose attached to its base and interconnecting phosphate groups. The minimization can be solved using classical optimization methods, such as the cyclic coordinate method that does not require derivatives [67]. In this context, each evaluation of the function builds a different ribose conformation. Consequently, the time complexity is proportional to the number of evaluations needed. However,

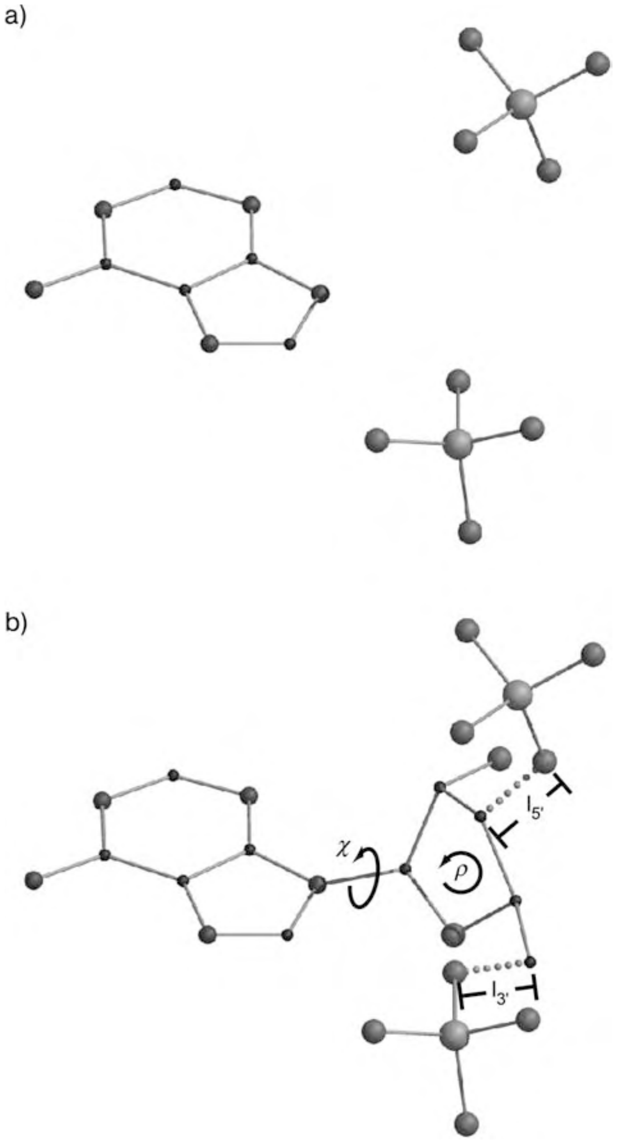


Figure 23 Ribose construction. (a) A base and two phosphate groups. (b) The ribose structure is appended to the base, and is parameterized by ρ and χ. The lengths of the implicit interconnections (shown in dashed lines), $l_{5'}$ and $l_{3'}$, respectively, represent the covalent C5′–O5′ and C3′–O3′ bonds, which quantify the precision of the construction.

we recently derived an optimal parameter estimation of Eq. (7), which solves the ribose optimization in constant time.

Theoretically, the backbone construction can be applied once all variables of the CSP are assigned values – when all bases and phosphate groups are in place. After the backbone construction, however, there is no guarantee that the final model is free from steric conflicts. Therefore, in practice, as soon as the two phosphate groups adjacent to a base are in place, the backbone is built for this nucleotide and the steric clashes verified. This, as in the former backtracking, allows us to prune the search tree.

4.2.2 **Probabilistic Backtracking**

Exhaustive searches for all valid RNA 3-D structures are useful to analyze possible alternative folding of an RNA. However, sometimes only one or few valid models are desired. We recently developed a probabilistic search algorithm that generates valid structures faster and with an increased diversity rate than that of the deterministic backtracking.

We select a random path from the root node of the search tree to any leaf. If at any variable assignment along the random path a constraint is not satisfied, a fixed-size regular backtracking is run until a consistent node is found, which resumes the probabilistic search. If the fixed-size regular backtracking cannot find a consistent node, then a new random path from the root node is selected.

4.2.3 **"Divide and Conquer"**

As for many problems, the "divide and conquer" paradigm has proven useful in RNA tertiary structure prediction as well. A "divide and conquer" algorithm splits a complex problem in many smaller and simpler to resolve subproblems. The solutions of the smaller problems are then combined in complete solutions of the larger problem.

RNA structures can be split into smaller fragments (as we saw in Section 3). Each fragment can be built independently and then merged in complete tertiary structures. In the CSP context, solving a fragment means locally enforcing the constraints in its scope: solving subset $W \subset V$ involves the verification of the constraints defined on W only. The solutions of W become the values of the domain of a new variable, say w, which is added to the CSP. The new CSP, CSP$'$, is defined over the variables $V' = (V \backslash W) \cup \{w\}$, the new domain for w and the new constraints $C' = C \backslash C_w$.

The search space sizes of CSP and CSP$'$ can be compared. Let $D = \{d_1, \ldots, d_n\}$ be the domains for the n variables in V and as defined in the CSP. $E = \{e_1, \ldots, e_k\}$ are the domains for the k variables in W, $E \subset D$. The domain for the new variable, w, is d_w, the solutions of W. If S and S' are, respectively, the search space sizes of CSP and CSP$'$, then from Eq. (4) we obtain:

$$S' < S \Leftrightarrow \frac{\prod_{i=1}^{n} d_i}{\prod_{i=1}^{k} e_i} \times d_w < \prod_{i=1}^{n} d_i \Leftrightarrow d_w < \prod_{i=1}^{k} e_i$$

showing the trivial result that CSP′ has a smaller search space than CSP if the search space size of CSP′ is larger than $|d_w|$, which is generally the case in the presence of actual structural constraints. To save even more time, when fragments of a tertiary structure correspond to RNA motifs, as described in Section 3, their solutions can directly be taken from the X-ray crystal structures [53]. We sometimes refer to this practice as RNA homology modeling.

4.3 MC-Sym at Work

Here we show how MC-Sym can be used to generate RNA tertiary structure models. The input description file has sections to describe the sequence, the nucleotide conformations, the nucleotide interactions, the constraints, and execution arguments.

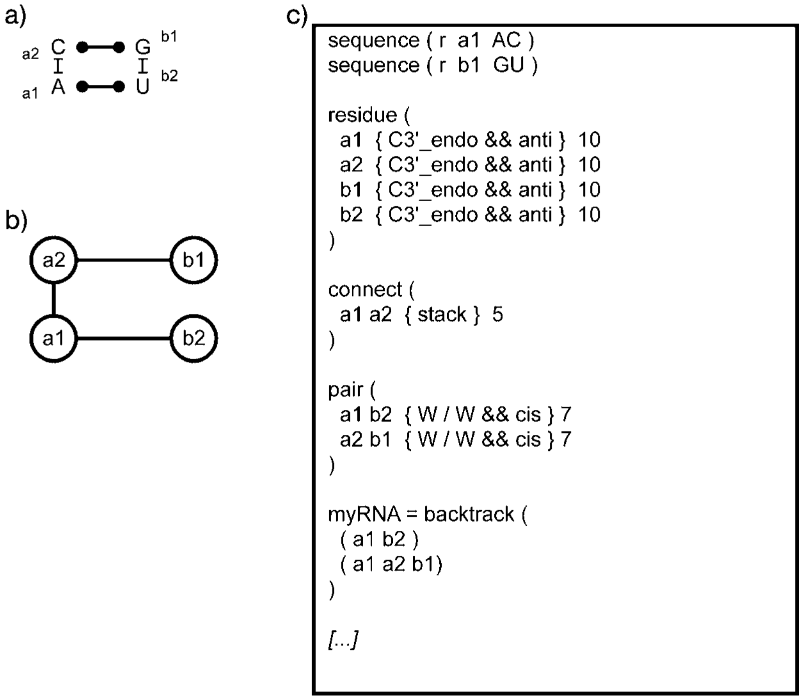
Figure 24 shows an example for the modeling of a tandem W/W base pairs: C=G stacked with A–U. Figure 24(a) shows the secondary structure of this simple fragment, made of two strands: "a" 5′-AC-3′ and "b" 5′-GU-3′. Figure 24(b) shows the spanning tree of the RNA graph, as chosen by the user, to build the fragment.

In the input (Figure 24c), the "sequence" section defines the two strands and introduces a global numbering system for the nucleotides. The "residue" section defines the nucleotide conformations, D_v, and sampling sizes; here, 10 different C3′-*endo anti* conformations. The nucleotide interactions are defined in the "connect" and "pair" statements, specified using the *LW*+ nomenclature. The "connect" statement is used for adjacent nucleotides in the sequence. In the example, one of the two base stack interactions is included in the spanning tree and five different stacking transformations will be assigned. The "pair" statement is used for the two Watson–Crick base pairs. Here, seven different Watson–Crick transformations will be assigned.

The "backtrack" statement defines the spanning tree, instructing MC-Sym about the order in which the nucleotides will be inserted in the models. Here, a1 is selected as the global referential, then b2 is Watson–Crick to a1, a2 is stacked with a1 and, finally, b1 is Watson–Crick to a2.

The domain specifications are translated by MC-Sym into queries to the appropriate nucleotide conformation or interaction database. The results of the logical queries define the domains. For instance, the query for the conformation of the cytosine a2 could match entries #5 and #8 in the conformation database shown in Table 2, resulting in the conformation domain $\{S_5, S_8\}$. Similarly, the Watson–Crick query for the a2–b1 interaction could match database entries #1, #3 and #8 in Table 3, resulting in the transformation domain $\{T_1, T_3, T_8\}$.

As indicated in Tables 2 and 3, the conformation and transformation domains come from X-ray crystal structures. The atomic coordinates for the



```
sequence ( r  a1  AC )
sequence ( r  b1  GU )

residue (
  a1  { C3'_endo && anti }  10
  a2  { C3'_endo && anti }  10
  b1  { C3'_endo && anti }  10
  b2  { C3'_endo && anti }  10
)

connect (
  a1 a2  { stack }  5
)

pair (
  a1 b2  { W / W && cis } 7
  a2 b1  { W / W && cis } 7
)

myRNA = backtrack (
  ( a1 b2 )
  ( a1 a2 b1)
)

[...]
```

Figure 24 Tandem $|W/W|$ base pairs. (a) RNA graph. (b) Spanning tree. (c) MC-Sym input. The "stack" keyword is used as a wildcard matching any of the four stacking types: upward, downward, inward or outward.

Table 2 *MC-Sym* nucleotide conformation database snippet

#	3-D coordinates set [a]	Symbols list	Origin [b]
1	S_1	A, C3′-*endo*, *anti*	1EVV ‘A’23
2	S_2	U, C3′-*endo*, *anti*	1FFK ‘0’55
3	S_3	G, C2′-*endo*, *anti*	1EHZ ‘A’18
4	S_4	A, C2′-*exo*, *anti*	1EVV ‘A’35
5	S_5	C, C3′-*endo*, *anti*	1JJ2 ‘0’361
6	S_6	U, C3′-*endo*, *syn*	1FFK ‘0’10
7	S_7	A, C4′-*exo*, *anti*	1JJ2 ‘0’407
8	S_8	C, C3′-*endo*, *anti*	1EHZ ‘A’13
[. . .]			

[a] Each S_i contains the 3-D coordinates of the nucleotide.
[b] PDB ID and numbering of the nucleotide.

conformations are directly extracted. The transformation between two nucleotides, A and B, is extracted by computing the relative transformation that places B's frame in A's local frame (Figure 25). If ${}^{O}M_A$ and ${}^{O}M_B$ are respectively the relative transformations that place the frame of A and B in

Table 3 *MC-Sym* nucleotide interaction database snippet

#	3-D coordinate set [a]	Symbol list	Origin [b]	
1	T_1	C-G, *Ws* / *Ww*, *cis*, antiparallel	1JJ2	'0'284, '0'367
2	T_2	U-C, adjacent, upward	1EHZ	'A'59, 'A'60
3	T_3	C-G, *Ww* / *Ww*, *cis*, antiparallel	1EVV	'A'27, 'A'43
4	T_4	A-C, adjacent	1FFK	'0'337, '0'338
5	T_5	A-C, outward	1EVV	'A'6, 'A'7
6	T_6	G-U, *Ww* / *Ws*, *cis*, parallel	1EHZ	'A'4, 'A'69
7	T_7	G-A, *Ss* / *Hh*, *cis*, antiparallel	1FFK	'0'2865, '0'2891
8	T_8	C-G, *Ww* / *Ww*, *cis*, antiparallel	1JJ2	'0'154, '0'182
[. . .]				

[a] Each T_i corresponds to the relative transformation matrix of the interaction.
[b] PDB ID and numbering of the two interacting nucleotides.

the global frame, O, then we extract the relative transformation ${}^{A}M_B$ so that ${}^{O}M_B = {}^{O}M_A \times {}^{A}M_B$. Isolating for ${}^{A}M_B$, we obtain ${}^{A}M_B = {}^{O}M_A^{-1} \times {}^{O}M_B$. We save the ${}^{A}M_B$ matrix in the database so that it can be reproduced for any other pair of bases in any local frame.

The MC-Sym database contains nearly 3000 nucleotide conformations and nearly 20 000 base interactions; hence, the domain size argument next to each conformation and interaction. It is the task of the modeler to assign domain sizes so that the conformational space of a given tertiary structure is correctly addressed – not too small to miss valid models and not too large to avoid prohibitive search space sizes.

4.3.1 Modeling a Yeast tRNA-Phe Stem–Loop

In Figure 26, we present a modeling example for the yeast tRNA-Phe T-stem–loop tertiary structure. The secondary structure of the stem–loop is shown in Figure 26(a). The stem and hairpin loop are modeled independently, and the results of each modeling merged. Figure 26(b–d) shows the three inputs. The first describes the structure of the first four base pairs of the stem. The second describes the hairpin loop, closed by the last base pair of the stem. The last merges the resulting fragments and, thus, models the entire stem–loop. Figure 26(e–g) illustrates the spanning trees defined by the three inputs.

The "res_clash" and "adjacency" statements parameterize the steric clashes and adjacency constraints, respectively. The "explore" statement launches the CSP solver. The RNA graph of the loop is divided into two sections by the W/H U54–A58 base pair, leaving the sequence adjacency implicit to the construction between G57 and A58, and between C60 and C61. Figure 26(h) shows one solution.

4.3.2 **Modeling a Pseudoknot**

To model the tertiary structure of the FIV pseudoknot (see Section 3.1.1), a novel methodological protocol based on mass spectroscopy and computer modeling was designed by Fabris and his coworkers. The experimental data were generated using multiplexing solvent-accessibility probes and chemical bifunctional crosslinkers with a characterization by an electrospray ionization Fourier transform mass spectrometry method (ESI-FTMS) [68]. The chemical and enzymatic probes cleave at specific sites or attack specific chemical groups that are exposed to the solvent. The secondary structure, detailed protection maps and inter-nucleotide distance information were then input to MC-Sym, which generated a set of consistent tertiary structures. Finally, the modeled structures were refined by energy minimization using the Crystallography and NMR System (CNS) [69].

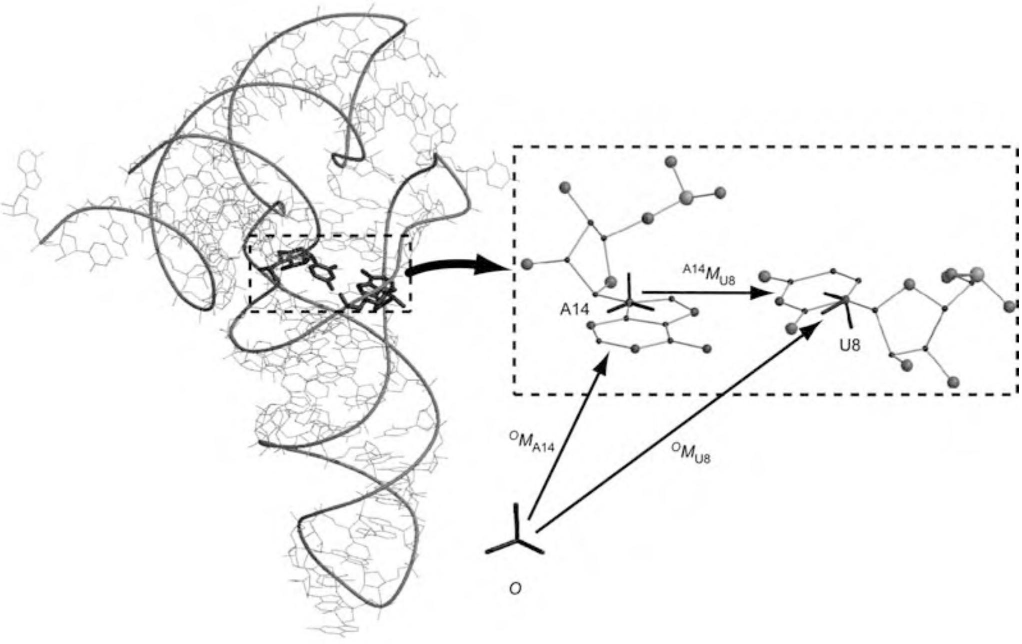
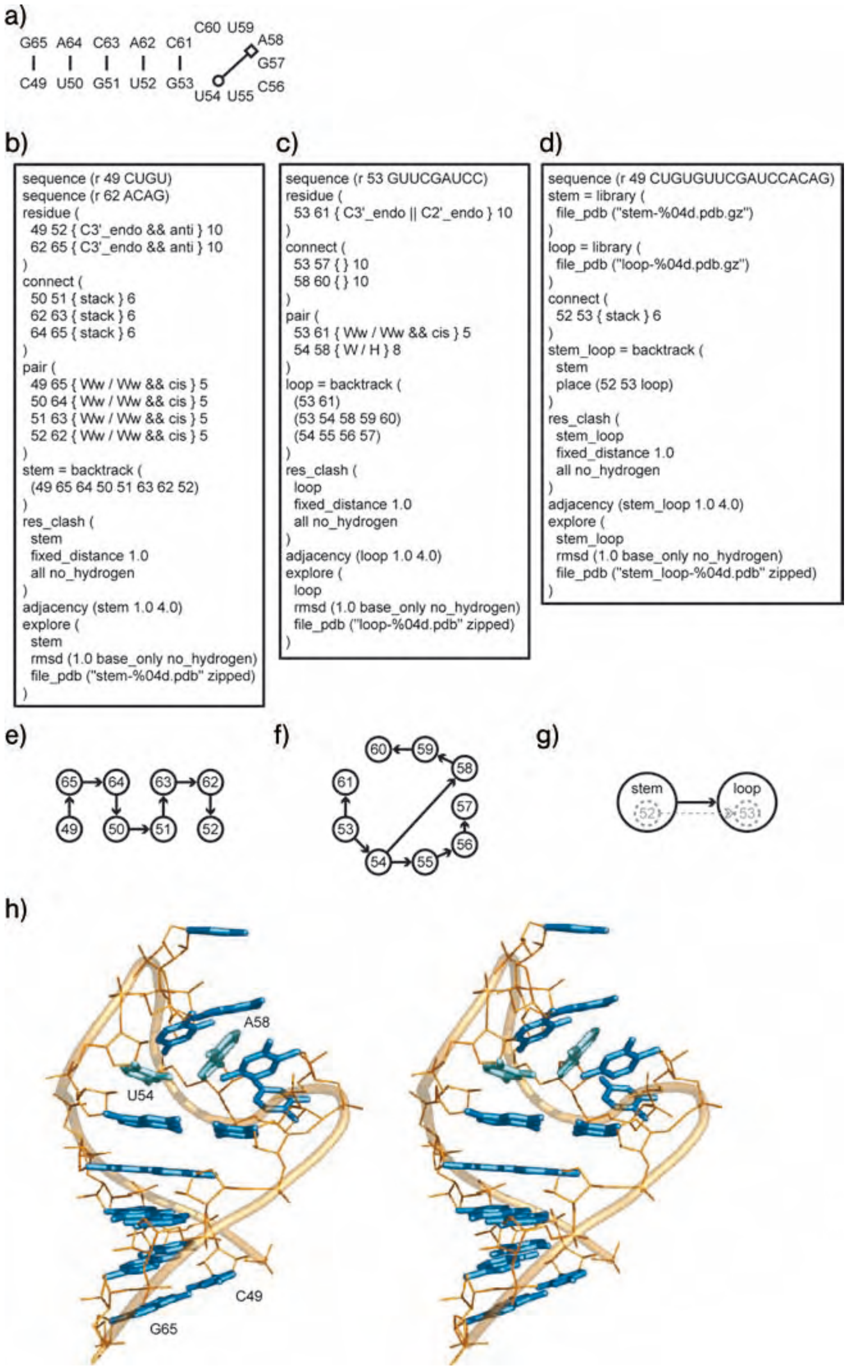


Figure 25 Extraction of the A14–U8 H/W base pair transformation in the yeast tRNA-Phe X-ray crystal structure (PDB ID 1EVV). A thread follows the strands by the phosphorus atoms. Hydrogen atoms are not shown. The base pair is zoomed and shown with the frames ($^{O}M_{A14}$ and $^{O}M_{U8}$) defined in the global frame, O. The transformation $^{A14}M_{U8} = {}^{O}M_{A14}^{-1} \times {}^{O}M_{U8}$ is extracted.



b)

```
sequence (r 49 CUGU)
sequence (r 62 ACAG)
residue (
 49 52 { C3'_endo && anti } 10
 62 65 { C3'_endo && anti } 10
)
connect (
 50 51 { stack } 6
 62 63 { stack } 6
 64 65 { stack } 6
)
pair (
 49 65 { Ww / Ww && cis } 5
 50 64 { Ww / Ww && cis } 5
 51 63 { Ww / Ww && cis } 5
 52 62 { Ww / Ww && cis } 5
)
stem = backtrack (
 (49 65 64 50 51 63 62 52)
)
res_clash (
 stem
 fixed_distance 1.0
 all no_hydrogen
)
adjacency (stem 1.0 4.0)
explore (
 stem
 rmsd (1.0 base_only no_hydrogen)
 file_pdb ("stem-%04d.pdb" zipped)
)
```

c)

```
sequence (r 53 GUUCGAUCC)
residue (
 53 61 { C3'_endo || C2'_endo } 10
)
connect (
 53 57 { } 10
 58 60 { } 10
)
pair (
 53 61 { Ww / Ww && cis } 5
 54 58 { W / H } 8
)
loop = backtrack (
 (53 61)
 (53 54 58 59 60)
 (54 55 56 57)
)
res_clash (
 loop
 fixed_distance 1.0
 all no_hydrogen
)
adjacency (loop 1.0 4.0)
explore (
 loop
 rmsd (1.0 base_only no_hydrogen)
 file_pdb ("loop-%04d.pdb" zipped)
)
```

d)

```
sequence (r 49 CUGUGUUCGAUCCACAG)
stem = library (
 file_pdb ("stem-%04d.pdb.gz")
)
loop = library (
 file_pdb ("loop-%04d.pdb.gz")
)
connect (
 52 53 { stack } 6
)
stem_loop = backtrack (
 stem
 place (52 53 loop)
)
res_clash (
 stem_loop
 fixed_distance 1.0
 all no_hydrogen
)
adjacency (stem_loop 1.0 4.0)
explore (
 stem_loop
 rmsd (1.0 base_only no_hydrogen)
 file_pdb ("stem_loop-%04d.pdb" zipped)
)
```

4.3.3 Cycles of Interactions

Traversal of a spanning tree to build tertiary structures implies a conceptual problem, as was pointed out in Section 4.2. A spanning tree does not cover all the arcs of the RNA graph. User constraints must be added to the input to make sure the dropped interactions are satisfied. However, such *ad hoc* constraints are difficult to define and compute. Lemieux and Major have designed a novel building approach to decompose an RNA graph in a series of minimum cycles of interactions [70], whose solutions can be combined by superimposing common arcs. The RNA graph in Figure 24, for instance, is a minimum cycle of four nucleotides. The product of the four transformation matrices is the identity matrix, representing an additional constraint to ensure the consistency of the cycle and the satisfaction of the four interactions.

These minimum interaction cycles could well be used as first-class objects in stochastic graph grammars [71, 72] to represent the tertiary structure of related RNAs and of their sequence alignment. This is similar, but yet more expressive, than stochastic context-free grammars employed to represent RNA secondary structures [73,74].

5 Perspectives

Accurate prediction of RNA tertiary structures from sequence alone is still an unresolved problem. In the meantime, formalizing RNA attributes, searching for higher-order levels of structural organization and modeling their tertiary structure represent current efforts towards better understanding of the RNA architectural principles. In addition, formalizing RNA structural knowledge in computer programs offers the possibility to apply it in a systematic and objective manner, allowing us to generate new and experimentally testable data.

The recent resolution of several RNA structures by X-ray crystallography, NMR, as well as by computer modeling, has allowed us to observe repeated RNA fragments and to infer their function. We are starting to understand the sequence constraints imposed by the tertiary structure of these fragments, and to discover local and global folding rules. In the coming years, as we can

←

Figure 26 Yeast tRNA T-stem–loop. (a) Secondary structure. (b) MC-Sym input for the stem fragment. (c) MC-Sym input for the loop fragment. (d) MC-Sym input for merging both fragments. (e) Spanning tree of (b). (f) Spanning tree of (c). (g) Spanning tree of (d). (h) Stereoview of a final model generated by MC-Sym. The bases are shown in blue; the U54–A58 W/H base pair in lighter blue. The backbone is shown in yellow. The thread follows the phosphodiester chain. Hydrogen atoms are not shown.

expect agreement on an RNA ontology (nomenclatures and formalisms), we might assist in the deployment and implementation of these folding rules into accurate RNA tertiary structure prediction algorithms. An RNA ontology will enable the interoperability of research results. Consequently, as we identify the RNAs of key cellular processes, determine their structure and characterize their role, we will be in a better position to manipulate them. As a result, we should observe an increase in the number and an improvement in the accuracy of RNA-based molecular medicine techniques.

References

1 Berman, H. M., J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne. 2000. The Protein Data Bank. Nucleic Acids Res. **28**: 235–42.

2 Gendron, P., S. Lemieux and F. Major. 2001. Quantitative analysis of nucleic acid three-dimensional structures. J. Mol. Biol. **308**: 919–36.

3 Yang, H., F. Jossinet, N. B. Leontis, L. Chen, J. Westbrook, H. M. Berman and E. Westhof. 2003. Tools for the automatic identification and classification of RNA base pairs. Nucleic Acids Res. **31**: 3450–60.

4 Altona, C. and M. Sundaralingam. 1972. Conformational analysis of the sugar ring in nucleosides and nucleotides. A new description using the concept of pseudorotation. J. Am. Chem. Soc. **94**: 8205–12.

5 Kraulis, P. J. 1991. MOLSCRIPT: a program to produce both detailed and schematic plots of protein structures. J. Appl. Crystallogr. **24**: 946–50.

6 Merritt, E. A. and M. E. Murphy. 1994. Raster3D version 2.0. A program for photorealistic molecular graphics. Acta Crystallogr. D **50**: 869–73.

7 Saenger, W. 1984. *Principles of Nucleic Acid Structure*. Springer, New York, NY.

8 Olson, W. K. 1980. Configurational statistics of polynucleotide chains. An updated virtual bond model to treat effects of base stacking. Macromolecules **13**: 721–8.

9 Major, F., M. Turcotte, D. Gautheret, G. Lapalme, E. Fillion and R. Cedergren. 1991. The combination of symbolic and numerical computation for three-dimensional modeling of RNA. Science **253**: 1255–60.

10 Gautheret, D., F. Major and R. Cedergren. 1993. Modeling the three-dimensional structure of RNA using discrete nucleotide conformational sets. J. Mol. Biol. **229**: 1049–64.

11 Paul, R. P. 1981. *Robot Manipulators: Mathematics, Programming and Control*. MIT Press, Cambridge, MA.

12 Duarte, C. M., L. M. Wadley and A. M. Pyle. 2003. RNA structure comparison, motif search and discovery using a reduced representation of RNA conformational space. Nucleic Acids Res. **31**: 4755–61.

13 Hershkovitz, E., E. Tannenbaum, S. B. Howerton, A. Sheth, A. Tannenbaum and L. D. Williams. 2003. Automated identification of RNA conformational motifs: theory and application to the HM LSU 23S rRNA. Nucleic Acids Res. **31**: 6249–57.

14 Murray, L. J. W., W. B. Arendall III, D. C. Richardson and J. S. Richardson. 2003. RNA backbone is rotameric. Proc. Natl Acad. Sci. USA **100**: 13904–9.

15 Schneider, B., Z. Morávek and H. M. Berman. 2004. RNA conformational classes. Nucleic Acids Res. **32**: 1666–7.

16 Serganov, A., Y. R. Yuan, O. Pikovskaya, et al. 2004. Structural basis

for discriminative regulation of gene expression by adenine- and guanine-sensing mRNAs. Chem. Biol. **11**: 1729–41.

17 Gabb, H. A., S. R. Sanghani, C. H. Robert and C. Prévost. 1996. Finding and visualizing nucleic acid base stacking. J. Mol. Graph. **14**: 6–11.

18 Leontis, N. B. and E. Westhof. 2001. Geometric nomenclature and classification of RNA base pairs. RNA **7**: 499–512.

19 Lemieux, S. and F. Major. 2002. RNA canonical and non-canonical base pairing types: a recognition method and complete repertoire. Nucleic Acids Res. **30**: 4250–63.

20 Lee, J. C. and R. R. Gutell. 2004. Diversity of base-pair conformations and their occurrence in rRNA structure and RNA structural motifs. J. Mol. Biol. **344**: 1225–49.

21 Leontis, N. B., J. Stombaug and E. Westhof. 2002. The non-Watson–Crick base pairs and their associated isostericity matrices. Nucleic Acids Res. **30**: 3479–531.

22 Walberer, B. J., A. C. Cheng and A. D. Frankel. 2003. Structural diversity and isomorphism of hydrogen-bonded base interactions in nucleic acids. J. Mol. Biol. **327**: 767–80.

23 Lescoute, A., N. B. Leontis, C. Massire and E. Westhof. 2005. Recurrent structural RNA motifs, isostericity matrices and sequence alignments. Nucleic Acids Res. **33**: 2395–409.

24 Ban, N., P. Nissen, J. Hansen, P. B. Moore and T. A. Steitz. 2000. The complete atomic structure of the large ribosomal subunit at 2.4 Å resolution. Science **289**: 905–20.

25 Wimberly, B. T., D. E. Brodersen, W. M. Clemons Jr., R. J. Morgan-Warren, A. P. Carter, C. Vonrhein, T. Hartsch and V. Ramakrishnan. 2000. Structure of the 30S ribosomal subunit. Nature **407**: 327–39.

26 Harms, J., F. Schluenzen, R. Zarivach, et al. 2001. High resolution structure of the large ribosomal subunit from a mesophilic eubacterium. Cell **107**: 679–88.

27 Leontis, N. B. and E. Westhof. 2003. Analysis of RNA motifs. Curr. Opin. Struct. Biol. **13**: 300–8.

28 Waugh, A., P. Gendron, R. Altman, et al. 2002. RNAML: a standard syntax for exchanging RNA information. RNA **8**: 707–17.

29 Jossinet, F. and E. Westhof. 2005. Sequence to structure (S2S): display, manipulate and interconnect RNA data from sequence to structure. Bioinformatics **1**: 3320–1.

30 Ullman, J. R. 1976. An algorithm for subgraph isomorphism. J. ACM **23**: 31–42.

31 Leontis, N. B., J. Stombaugh and E. Westhof. 2002. Motif prediction in ribosomal RNAs. Lessons and prospects for automated motif prediction in homologous RNA molecules. Biochimie **84**: 961–73.

32 Nagaswamy, U. and G. E. Fox. 2002. Frequent occurrence of the T-loop RNA folding motif in ribosomal RNAs. RNA **8**: 1112–9.

33 Nissen, P., J. A. Ippolito, N. Ban, P. B. Moore and T. A. Steitz. 2001. RNA tertiary interactions in the large ribosomal subunit: the A-minor motif. Proc. Natl Acad. Sci. USA **98**: 4899–903.

34 Ogle, J. M., F. V. Murphy IV, M. J. Tarry and V. Ramakrishnan. 2002. Selection of tRNA by the ribosome requires a transition from an open to a closed form. Cell **111**: 721–32.

35 Klein, D. J., T. M. Schmeing, P. B. Moore and T. A. Steitz. 2001. The kink-turn: a new RNA secondary structure motif. EMBO J. **20**: 4214–21.

36 Pley, H. W., K. M. Flaherty and D. B. McKay. 1994. Model for an RNA tertiary interaction from the structure of an intermolecular complex between a GAAA tetraloop and an RNA helix. Nature **372**: 111–3.

37 Jaeger, L., F. Michel and E. Westhof. 1994. Involvement of a GNRA tetraloop in long-range RNA tertiary interactions. J. Mol. Biol. **236**: 1271–6.

38 Bélanger, F., M. G. Gagnon, S. V. Steinberg, P. R. Cunningham and L. Brakier-Gingras. 2004. Study of the functional interaction of the 900 tetraloop of 16S ribosomal RNA with helix 24 within the bacterial ribosome. J. Mol. Biol. **338**: 683–93.

39 TEN DAM, E., C. W. A. PLEIJ AND D. E. DRAPER. 1992. Structural and functional aspects of RNA pseudoknots. Biochemistry **47**: 11665–76.

40 GIEDROC, D. P., C. A. THEIMER AND P. L. NIXON. 2000. Structure, stability and function of RNA pseudoknots involved in stimulating ribosomal frameshifting. J. Mol. Biol. **298**: 167–85.

41 MORIKAWA, S. AND D. H. L. BISHOP. 1992. Identification and analysis of the *gap–pol* ribosomal frameshift site of feline immunodeficiency virus. Virology **186**: 389–97.

42 YU, E. T., Q. ZHANG AND D. FABRIS. 2005. Untying the FIV frameshifting pseudoknot structure by MS3D. J. Mol. Biol. **345**: 69–80.

43 RUFFNER, D. E., G. D. STORMO AND O. C. UHLENBECK. 1990. Sequence requirements of the hammerhead RNA self-cleavage reaction. Biochemistry **29**: 10695–702.

44 PLEY, H. W., K. M. FLAHERTY AND D. B. MCKAY. 1994. Three-dimensional structure of a hammerhead ribozyme. Nature **372**: 68–74.

45 SCOTT, W. G., J. T. FINCH AND A. KLUG. 1995. The crystal structure of an all-RNA hammerhead ribozyme: a proposed mechanism for RNA catalytic cleavage. Cell **81**: 991–1002.

46 LEGAULT, P., C. G. HOOGSTRATEN, E. METLITZKY AND A. PARDI. 1998. Order, dynamics and metal-binding in the lead-dependent ribozyme. J. Mol. Biol. **284**: 325–35.

47 LEMIEUX, S, P. CHARTRAND, R. CEDERGREN AND F. MAJOR. 1998. Modeling active RNA structures using the intersection of conformational space: application to the lead-activated ribozyme. RNA **4**: 739–49.

48 WEDEKIND, J. E. AND D. B. MCKAY. 1999. Crystal structure of a lead-dependent ribozyme revealing metal binding sites relevant to catalysis. Nat. Struct. Biol. **6**: 261–8.

49 DAVID, L., D. LAMBERT, P. GENDRON AND F. MAJOR. 2001. Leadzyme. Methods Enzymol. **341**: 518–40.

50 PINARD, R., D. LAMBERT, J. E. HECKMAN, et al. 2001. The hairpin ribozyme substrate binding domain: a highly constrained D-shaped conformation. J. Mol. Biol. **307**: 51–65.

51 PINARD, R., D. LAMBERT, G. POTHIAWALA, F. MAJOR AND J. M. BURKE. 2004. Modifications and deletions of helices within the hairpin ribozyme–substrate complex: an active ribozyme lacking helix 1. RNA **10**: 395–402.

52 LESCOUTE, A. AND E. WESTHOF. 2006. Topology of the three-way junctions in folded RNAs. RNA **12**: 83–93.

53 KHVOROVA, A., A. LESCOUTE, E. WESTHOF AND S. D. JAYASENA. 2003. Sequence elements outside the hammerhead ribozyme catalytic core enable intracellular activity. Nat. Struct. Biol. **10**: 708–12.

54 MASSIRE, C. AND E. WESTHOF. 1998. MANIP: an interactive tool for modeling RNA. J Mol. Graph. Model. **16**: 197–205.

55 HILEY, S. L. AND R. A. COLLINS. 2001. Rapid formation of a solvent-inaccessible core in the *Neurospora* Varkud satellite ribozyme. EMBO J. **20**: 5461–9.

56 HOFFMANN, B., G. T. MITCHELL, P. GENDRON, F. MAJOR, A. A. ANDERSEN, R. A. COLLINS AND P. LEGAULT. 2003. NMR structure of the active conformation of the Varkud satellite ribozyme cleavage site. Proc. Natl Acad. Sci. USA **100**: 7003–8.

57 TAMURA, M. AND S. R. HOLBROOK. 2002. Sequence and structural conservation in RNA riboses. J. Mol. Biol. **320**: 455–74.

58 OLIVIER, C., G. POIRIER, P. GENDRON, A. BOISGONTIER, F. MAJOR AND P. CHARTRAND. 2005. Identification of a conserved RNA motif essential for She2p recognition and mRNA localization to the yeast bud. Mol. Cell. Biol. **25**: 4752–766.

59 CHARTRAND, P., X.-H. MENG, R. H. SINGER AND R. M. LONG. 1999. Structural elements required for the localization of *ASH1* mRNA and of a green fluorescent protein reporter particle *in vivo*. Curr. Biol. **9**: 333–6.

60 LONG, R. M., W. GU, E. LORIMER, R. H. SINGER AND P. CHARTRAND. 2000. She2p is a novel RNA-binding protein that recruits the Myo4p–She3p complex to *ASH1* mRNA. EMBO J **19**: 6592–601.

61 MALHOTRA, A., R. K. TAN AND S. C. HARVEY. 1990. Prediction of the three-dimensional structure of *Escherichia coli* 30S ribosomal subunit: a molecular mechanics approach. Proc. Natl Acad. Sci. USA **87**: 1950–4.

62 MAJOR, F. 2003. Building three-dimensional ribonucleic acid structures. IEEE Comput. Sci. Eng. **Sep/Oct**: 44–53.

63 HENTENRYCK, P. V. 1989. *Constraint Satisfaction in Logic Programming*. MIT Press, Cambridge, MA.

64 DECHTER, R. AND D. FROST. 2002. Backjump-based backtracking for constraint satisfaction problems. Artificial Intell. **136**: 147–88.

65 MAJOR, F., D. GAUTHERET AND R. CEDERGREN. 1993. Reproducing the three-dimensional structure of a tRNA molecule from structural constraints. Proc. Natl Acad. Sci. USA **90**: 9408–12.

66 LEMIEUX, S., S. OLDZIEJ AND F. MAJOR. 1998. Nucleic acids: qualitative modeling. In SCHLEYER, P. V. R., N. L. ALLINGER, T. CLARK, et al. (eds.), *Encyclopedia of Computational Chemistry*. Wiley, Chichester: 000–00.

67 BAZARAA, M. S. AND C. M. SHETTY. 1979. *Nonlinear Programming Theory and Algorithms*. Wiley, New York, NY.

68 YU, E. T. AND D. FABRIS. 2003. Direct probing of RNA structures and RNA-protein interactions in the HIV-1 packaging signal by chemical modification and electrospray ionization Fourier transform mass spectrometry. J. Mol. Biol. **330**: 211–23.

69 BRÜNGER, A.T., P. D. ADAMS, G. M. CLORE, et al. 1998. Crystallography & NMR system: a new software suite for macromolecular structure determination. Acta Crystallogr. D **54**: 905–21.

70 LEMIEUX, S. AND F. MAJOR. 2006. Automated extraction and classification of RNA tertiary structure cyclic motifs. Nucl. Acids Res. **34**: 2340–6.

71 NAGL, M. 1987. Set theoretic approaches to graph grammars. In EHRIG, H., M. NAGL, G. ROZENBERG AND A. ROSENFELD (eds.), *Graph-grammars and their Application to Computer Science*. Springer, Berlin: 41–54.

72 JONES, C. V. 1993. An integrated modeling environment based on attributed graphs and graph-grammars. Decision Support Syst. **10**: 255–75.

73 SAKAKIBARA, Y., M. BROEN, R. HUGHEY, I. S. MIAN, K. SJÖLANDER, R. C. UNDERWOOD AND D. HAUSSLER. 1994. Stochastic context-free grammars for tRNA modeling. Nucleic Acids Res. **22**: 5112–20.

74 DOWELL, R. D. AND S. R. EDDY. 2004. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. BMC Bioinformatics **5**: 71.

Part 5 Analysis of Molecular Interactions

16
Docking and Scoring for Structure-based Drug Design

Matthias Rarey, Jörg Degen and Ingo Reulecke

1 Introduction

As a result of structural genomics activities, more and more protein structures are becoming available weekly (see also Chapter 13). At the beginning of 2005, more than 29 000 protein structures were deposited in the Protein Data Bank (PDB) [25, 26]. Although experimental structure elucidation via X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy cannot be fully automated at present, it has become a routine task for several protein structural classes. If one excludes complex structures like membrane-bound proteins (e.g. ion channels, G-protein-coupled receptors), one can expect that the three-dimensional (3-D) structure for a protein of interest can be solved with a reasonable chance.

In order to understand a protein's function – or even more challenging to modulate a protein's function – it is important to be able to predict and understand the interactions a protein undergoes with other biomolecules. On the computational side, one has to solve the docking problem: given two molecules, we would like to know whether these molecules will form a complex, how stable this complex is and what it looks like geometrically. Interactions occur between all types of biomolecules – proteins form stable protein–protein complexes, and interact with RNA, DNA and small organic molecules. The interactions between small organic molecules and proteins are of special interest. These interactions often give important hints on the protein's function. Moreover, small molecules can inhibit or activate protein functions and, therefore, play a dominant role in pharmaceutical research. Recently, small molecules have also been applied as a biological tool for revealing

Bioinformatics – From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

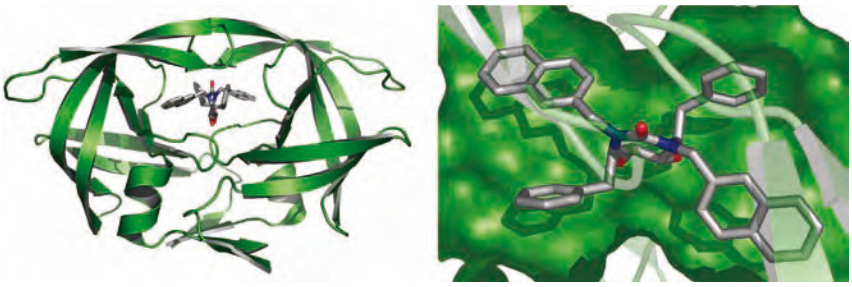


Figure 1 Protein–ligand complex. HIV protease in complex with the nonpeptidic inhibitor XK263 from Dupont Merck.

protein function [276]. The focus of this chapter is on reviewing computational methods for predicting interactions between proteins and small molecules, called *ligands* in this context. Chapter 17 concentrates on modeling protein interactions with macromolecular binding partners.

Figure 1 gives an example of a protein–ligand complex. The protein is HIV-protease, a protein from the human immunodeficiency virus (HIV) with the function of cutting a translated multi-protein peptide chain into pieces which fold to HIV proteins (see also Chapter 40). Since this function is not required by the host, i.e. the infected patient, HIV protease is a useful drug target: inhibiting HIV protease prevents the virus from replicating. The ligand molecule is a known inhibitor of HIV protease. The two molecules show a perfect complementarity of the molecule's shapes. In Section 3 we will see that steric as well as physicochemical complementarity is a key aspect in molecular docking calculations.

A computer method for the prediction of protein–ligand complexes has two major ingredients: an algorithm creating potential protein–ligand complex geometries (*pose generation*) and a function predicting the binding affinity of the ligand to the protein based on the complex geometry (*scoring*). Scoring and pose generation both contribute to the difficulty of the docking problem. Several physical effects like electrostatics, van der Waals forces, hydrogen bonding, and hydrophobic and entropic effects influence the binding affinity, some of which can only be estimated roughly. Since these effects can increase or decrease binding affinity, the balance of the terms is of major importance and makes scoring such a difficult task. An introduction to scoring functions will be given in Section 2.

Pose generation involves consideration of several degrees of freedom. The most important ones are the relative orientation of the two molecules and the conformation of the ligand molecule. Apart from these, the protein conformation may also vary, water molecules can be located at the interface between the molecules and the protonation states of the molecules can change. All of these variables are subject to a complicated network of constraints. A small change in one variable may decide whether an overlap between the two molecules occurs or whether a hydrogen bond between two groups can be formed. This implies that the scoring functions for protein–ligand docking typically contain several local minima, and have steep slopes or points of degeneracy and discontinuity. Thus, they are difficult to optimize. Several pose generation methods will be discussed in Section 3.

Although no general solution of the docking problem is available today, several scoring functions and pose generation algorithms for various variants of the docking problem have been developed and successfully applied. This chapter will give an overview of these functions and algorithms. Additional recent review articles on software for structure-based drug design can be found in the literature [42, 110, 115, 152, 168, 187, 209, 243, 265]. Before starting our methods overview, we will summarize the types of docking problems and describe some typical application scenarios.

1.1 A Taxonomy of Docking Problems

Docking problems can best be classified by the type of input molecules. In macromolecular docking, two macromolecules like proteins or DNA are docked. The major characteristics of these problems are that the complex typically has a large contact area and that the molecules, although still flexible, have a fixed overall shape. These features imply that methods based on geometric properties such as shape complementarity alone can already be used efficiently for creating energetically favorable complexes. A survey of methods for protein–protein docking is given in Chapter 17 and elsewhere [197, 275].

The second class contains docking problems in which a small molecule is docked to a macromolecule. The macromolecule can be a DNA fragment or a protein; in the latter case, the problem is called protein–ligand docking. In general, the small molecule is an organic molecule, e.g. a natural substrate or an inhibitor like the protease inhibitor mentioned in the example above.

Small-molecule docking differs substantially from macromolecular docking in the fact that the ligand is typically not fixed in its overall shape. The conformational flexibility of the ligand molecule is of importance and geometric properties alone are often not sufficient to determine low-energy complexes. Even in cases in which the molecule is more or less rigid, the shape of the molecule is not characteristic enough to find low-energy complexes based on shape alone. Therefore, the algorithms that have been developed for small-molecule docking differ from those for macromolecular docking.

For various reasons pertaining to important drug properties, like their bioavailability, most drugs are small molecules. Therefore, small-molecule docking is of great interest in pharmaceutical research. Since pharmaceutical targets are often proteins, most docking algorithms are developed for protein–ligand docking.

In principle, docking of small molecules to DNA can be handled by the same algorithms. Differences occur in the handling of DNA-specific binding mechanisms like covalent binding or binding of so-called intercalators [235]. An intercalator is a small molecule that binds to DNA between two subsequent bases. Since a structural change in the DNA is necessary for binding, conformational flexibility of the DNA must be taken into account. This can be done by docking into a distorted DNA structure [112]. A better alternative is to deal with the DNA flexibility directly. For example, Zacharias and coworkers [305] developed an algorithm based on normal mode analysis for handling structural changes of the DNA during docking calculations.

Protein–ligand docking problems can be further distinguished by the size of the ligand molecule. The typical ligand molecule occurring in drug design docking problems has about five to 12 rotatable bonds. Thus, often the question of placing only a part of the ligand, a fragment, in the active site of the protein arises. In contrast to a typical drug molecule, the conformational space of a fragment is quite small and does not necessarily have to be taken into account explicitly. Therefore, algorithms handling only the relative orientation of the two molecules (rigid-body docking algorithms) can be applied. Docking fragments to proteins is a subproblem handled in several more complex docking algorithms. Some examples are the protein–ligand docking algorithms in which flexibility is handled by dividing the ligand into smaller fragments (see Section 3.2.2), *de novo* design algorithms in which new molecules are created from a fragment database (see Section 5.2) or combinatorial docking algorithms in which combinatorial libraries of molecules are docked by combining placements for individual building blocks of the library (see Section 4.2).

Orthogonal to the classification by the type of input molecules, the second important parameter for categorizing docking algorithms is the time spent per prediction. The number of molecules which have to be processed covers a range from single molecules to several hundred thousands and depends on the specific application scenario.

1.2 Application Scenarios in Structure-based Molecular Design

Molecular docking techniques are frequently used during the lead generation and optimization phase of a drug design project. Most prominent is surely the search for new lead structures in the very early phase of drug design. If the structure of a target protein of interest is available, molecular docking can be used to prioritize compounds in a virtual library [187]. This process, called *structure-based virtual screening*, simply docks each compound of the library into the active site of the target protein and estimates the free binding energy using the scoring function. Virtual screening has a quite long history. Only recently, however, have the methods become applicable due to better docking algorithms and scoring functions as well as fast and cheap computer hardware.

Even with modern scoring functions, virtual screening gives only rough estimates for binding affinities, and is therefore a prefilter used to reduce the number of compounds entering the costly and time-consuming experimental validation. Nevertheless, several case studies have been published in which virtual screening was used to select up to a few hundred compounds which were then experimentally tested [164]. All case studies have two aspects in common: (i) substantial knowledge of the modeler went into the virtual screening process with the consequence that we cannot consider the process as fully automated and (ii) in all cases the authors were able to identify reasonable lead structures. In summary, we can say that although virtual screening is not a fully automated process, it is a very powerful tool in the hands of experienced modelers.

Structure-based virtual screening is often combined with experimental high-throughput screening. For example, virtual screening can be used to preselect promising sublibraries from the compound inventory. Virtual screening may be less reliable than experimental high-throughput screening; however, it gives results which are much better suited for in-depth analysis. Since the proposed complex structure is available, the modeler is able to judge on the basis of this structure, cluster compounds by their binding mode and further reduce the set of solutions by filtering out unwanted geometrical solutions.

The ability to quickly access potential complex geometries makes molecular docking an essential tool for structure-guided lead optimization. Based on a validated lead, focused libraries can be designed and evaluated before the library has to be synthesized. Individual modifications for improving the binding affinity can be suggested much more easily if the complex structure is available. Functional groups in the molecule that are not involved in complex formation can be identified on the basis of the complex geometry. These groups are of major interest in the optimization of second-order drug properties like bioavailability, specificity and synthesizability.

In the near future, we can expect that more and more structures of functionally important human proteins will become available. This will open a new application route for molecular docking tools. With so-called *inverse screening*, we can predict the binding affinity of a lead compound to a large set of protein structures. With such an approach, further proteins interacting with the lead apart from the target protein can be identified. This will give important hints to potential side-effects, toxicity or inappropriate absorption, distribution, metabolism and elimination (ADME) profile of the lead. First publications show that molecular docking can, in principle, be applied in this scenario [223].

In the virtual world we are not limited to existing, already synthesized molecules. Molecular docking can be applied to hypothetical molecules, opening up a completely new route in drug design. In the 1980s, the idea arose of constructing a molecule inside the active site of a protein from scratch (*de novo* ligand design). Today, *de novo* design is an accepted alternative strategy for lead identification, especially if the protein structure is available. The main issue in modern *de novo* design methods is to estimate the synthetic accessibility of a compound. Section 5 summarizes the current strategies for taking synthetic accessibility into account.

2 Scoring Protein–Ligand Complexes

As virtual high-throughput screening is gaining importance in the lead discovery process, the need for a reliable scoring function becomes ever more pressing. Scoring functions have to fulfill three kinds of requirements in docking applications – different poses of a ligand inside a protein binding site have to be compared in order to identify the natural binding mode. Many ligands have to be ranked according to their binding affinity in order to distinguish possible lead structures from nonbinding molecules. Thus, as a scoring function needs to be evaluated very often, it has to be very fast to compute.

Since scoring is such a crucial concern in virtual drug design, a number of review articles deal with this problem in great detail, most recently Refs. [40, 103, 151, 161, 262]. In the following we will give a short overview over the basic features of protein–ligand association and commonly used scoring functions.

2.1 Modeling Protein–Ligand Interactions

A noncovalently bound inhibitor is subject to the dynamic equilibrium between the complex R–L and the uncomplexed state where both the ligand L and the receptor (protein) R are surrounded by solvent molecules (Figure 2). Applying the law of mass action, one can derive a measure of binding free energy from each species' concentration and chemical activity, respectively. Conversely, if we know the binding free energy of a ligand with regard to a certain target protein, we also know if it will bind to the protein or prefers the solvated state. Equation (1) reveals the relationship between the binding-free energy $\Delta G_{\text{binding}}$ and the individual concentrations, which are denoted by the

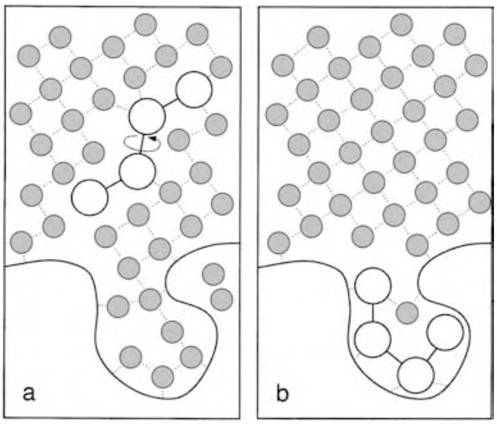


Figure 2 Energetic contributions. Schematic representation of (a) receptor R and ligand L uncomplexed and both surrounded by solvent molecules (shaded circles), and (b) receptor–ligand complex R–L. Hydrogen bonds are represented by dotted lines.

bracket terms.

$$\Delta G_{\text{binding}} = \Delta H_{\text{binding}} - T\Delta S_{\text{binding}} = -RT \ln \frac{[R-L]}{[R] \cdot [L]} , \tag{1}$$

where R is the gas constant and T is the absolute temperature. Binding free energy is composed of enthalpic and entropic contributions: Binding entropy $\Delta S_{\text{binding}}$ is a term derived from a change in the degree of order upon complex formation due to the change in the number of translational, rotational and conformational degrees of freedom, whereas the binding enthalpy $\Delta H_{\text{binding}}$ sums the changes of inter- and intramolecular forces. Typical attractive intermolecular forces are of electrostatic nature, but induction and dispersion effects also play a crucial role. With decreasing distance between the atoms, intermolecular forces become repulsive due to the penetration of the electron shells. A special case of intermolecular force is hydrogen bonding which occurs between a hydrogen atom bound to an electronegative atom and the lone pair of another electronegative atom [44]. Electronegative atoms relevant for protein–ligand docking are mainly oxygen and nitrogen.

When estimating binding free energy, one should also keep in mind that the protein, the ligand and the complex are surrounded by solvent molecules, which also take part in interactions. In particular, water, which is the natural solvent for a protein, has properties which are difficult to take into account [302].

2.2 Scoring Functions based on Force Fields

Force field or first principle type scoring approaches are of the general form

$$\begin{aligned} E_{\text{total}} &= \sum_{\text{bond}} K_r (r - r_{\text{eq}})^2 + \sum_{\text{angles}} K_\theta (\theta - \theta_{\text{eq}})^2 + \sum_{\text{dihedrals}} \frac{V_n}{2} [1 + \cos(n\phi - \gamma)] \\ &\quad + \sum_{i<j} \left[\frac{A_{ij}}{R_{ij}^{12}} - \frac{B_{ij}}{R_{ij}^{6}} + \frac{q_i q_j}{\varepsilon R_{ij}} \right], \end{aligned} \qquad (2)$$

where the energy E_{total} of a system is computed by summing up over inner strain and (noncovalent) binding energy values. Energies resulting from deviations from standard bond lengths r_{eq} and angles θ_{eq} are calculated by simple harmonic potential functions. Out-of-plane deviations ϕ are penalized using a periodic function, and intermolecular forces between the atoms i and j are approximated by Lennard–Jones and Coulomb expressions, which depend on the distance R_{ij} between the atoms. Force field calculations require many parameters which are optimized for specific types of molecular systems. Force fields often used for biomolecular interactions are AMBER [64], OPLS-AA [139], CHARMM [188] and GROMOS [217].

As can be seen in Eq. (2), force field expressions are lacking terms for entropic contributions. Therefore, force fields are mainly used in molecular dynamics (MD) or Monte Carlo (MC) methods. In these methods, free energy and entropy, respectively, are derived from generating an ensemble of states and applying the Boltzmann equation [27]. Solvation effects can be considered in principle, either indirectly by using a distance-dependent dielectric constant (Poisson–Boltzmann model) for simulating electrostatic shielding [130] or by considering the solvent molecules explicitly.

Such simulations can be performed in order to find the natural binding mode and the free energy of the system. In order to rank two compounds against each other a common reference state is required. Figure 3 shows the thermodynamic cycle for calculating the relative binding free energy $\Delta\Delta G_{L \to L*}$ of a ligand L* with respect to another ligand L for a common receptor R. ΔG_3 and ΔG_4 denote the binding free energy of L and L* with respect to R and $\Delta\Delta G_{L \to L*}$ the difference between them. In a thermodynamic cycle the energy is independent from the path it was derived from and thus, if ΔG_3 is known from experiment, ΔG_4 can be calculated from ΔG_1 and ΔG_2. These are the energies needed for transforming L to L* uncomplexed and in complex with R, respectively. They can be computed from stepwise thermodynamic integration over the reaction pathway. In practice, the transformation between the two systems is replaced by a series of transformations between non-physical intermediate states. This method is called free energy perturbation (FEP) (see Ref. [156] for further details).

$$\begin{tikzcd}
\mathbf{R + L} \arrow[r, "\Delta \mathbf{G}_1"] \arrow[d, "\Delta \mathbf{G}_3"'] & \mathbf{R + L^*} \arrow[d, "\Delta \mathbf{G}_4"] \\
\mathbf{R{-}L} \arrow[r, "\Delta \mathbf{G}_2"'] & \mathbf{R{-}L^*}
\end{tikzcd}$$

Figure 3 Thermodynamic cycle. Thermodynamic cycle for calculating the relative binding free energy difference $\Delta\Delta G_{\text{binding}}$ between two ligands L and L^* binding to the same protein R.

$$\Delta\Delta G_{L\rightarrow L^*} = \Delta G_4 - \Delta G_3 = \Delta G_2 - \Delta G_1 \; . \tag{3}$$

It is also possible to calculate the total binding free energy ΔG_4 of L^* by using a "dummy molecule" instead of L. The dummy molecule has no energetic influence on the complex, therefore ΔG_3 is zero. Since ΔG_3 is zero, ΔG_4 is derived directly from the thermodynamic calculations. As this FEP method requires many simulation steps all over the reaction pathway to acquire ensemble averages and according free energy values, it is very time-consuming and not applicable for screening purposes.

Åqvist and coworkers [12] introduced the linear interaction energy method (LIE), by which simulations are carried out only for the corners of the thermodynamic cycle. Rather than integrating over the reaction pathway, binding free energy is linearly approximated depending on the difference in polar and non-polar interaction energy before and after complex formation. The polar and nonpolar free energy contributions are estimated according to:

$$\Delta G_{\text{bind}} = \alpha\Delta \left\langle V_{l-s}^{\text{vdw}} \right\rangle + \beta\Delta\langle V_{l-s}^{\text{el}}\rangle + \gamma \, , \tag{4}$$

where the $\Delta\langle V_{l-s}\rangle$ terms denote the differences in the MD or MC averages between the nonbonded and the complexed state, separated in van der Waals (vdw) and electrostatic (el) interactions of the ligand with the surrounding environment. α, β and γ are constants.

Another method for speeding up force field-based docking calculations is to precalculate energy values for different kinds of atom probes placed at the nodes of a grid which is superposed with the protein's active site. Computing pairwise energies between ligand and protein atoms is therefore reduced to calculations pertaining to the grid points near to the ligand atoms. The historically first and still most frequently used software for this task is the GRID program [106]. Recently, Pearlman [226] introduced the OWFEG (one-window free energy grid) method, by which grid-point energies are derived from FEP calculations and directly related to free energy values.

2.3 Empirical Scoring

Empirical scoring functions are based on correlating geometric parameters of the protein–ligand complex with measured binding free energies. The overall score is a sum of terms representing different physicochemical effects contributing to both entropic and enthalpic energy changes on binding. This approach was pioneered by Böhm [34]. He applied such formula to protein–ligand complexes in the *de novo* design program LUDI [33,35]:

$$\begin{aligned}\Delta G_{\text{binding}} &= \Delta G_0 + \Delta G_{hb} \sum_{\text{H-bonds}} f(\Delta R, \Delta\alpha) + \Delta G_{\text{ionic}} \sum_{\text{ionic int.}} f(\Delta R, \Delta\alpha) \\ &\quad + \Delta G_{\text{lipo}} |A_{\text{lipo}}| + \Delta G_{\text{rot}} NROT \ . \end{aligned} \tag{5}$$

In LUDI, the binding free energy $\Delta G_{\text{binding}}$ is calculated by counting hydrogen bonds, ionic interactions, the lipophilic contact surface area between ligand and receptor, and the number of rotatable bonds in the ligand. Hydrogen bonds and ionic interactions, which diverge from ideal geometry by distance ΔR or angle $\Delta\alpha$, score with a linearly decreasing value. The contact surface is estimated for modeling the hydrophobic effect and the number of rotatable bonds is used to rate the loss of entropy due to freezing the ligand in a specific conformation upon binding. The individual scoring parameters ΔG_i originate from multiple linear regression analysis of 45 receptor–ligand complexes from the PDB with experimental determined binding affinities. The intercept ΔG_0 is regarded as entropic contribution due to the loss of the ligand's degrees of freedom of rotation and translation.

Based on Böhm's work many empirical scoring functions were released which differ mainly in the functional form for the individual terms [37,77,87, 92,99,121,134,148,159,205,242,249,272,279,293,294]. For example, pairwise contact terms or atom-based partial $\log P_{o/w}$ parameters are used instead of surface area terms for incorporating the hydrophobic effect. The term for loss of conformational entropy is also implemented in different ways, accounting for the fact that not all of the ligand's side-chains get stuck in a specific conformation and the rotamers are not independent from each other. Furthermore, with the increasing number of elucidated 3-D structures the parameterization datasets became more voluminous and other regression techniques (partial least squares [93], genetic algorithms [105], neural networks [201]) have been introduced that seem more appropriate for this kind of problem. POEM [8] is a recently published method which combines the design of experiment (DOE) approach [203] with various regressing techniques in order to train an empirical scoring function on a specific target protein.

2.4 Knowledge-based Scoring

Apart from the empirical scoring functions, the so-called knowledge-based approaches or potentials of mean force, originating from protein folding studies [268], have become quite popular. Assuming that an observed crystallographic complex represents the optimum placement of the ligand relative to the protein, protein–ligand atom pair potentials can be derived from the distance distributions of interactions between atoms of specific types found in structural databases. A popular member of this class of scoring functions is the PMF function, introduced by Muegge [208]:

$$A_{ij}(r) = -k_{\mathrm{B}} T \ln \left[f^{j}_{\mathrm{Vol\_corr}}(r) \frac{\rho^{ij}_{\mathrm{seg}}(r)}{\rho^{ij}_{\mathrm{bulk}}} \right] . \tag{6}$$

The potential of mean force A_{ij} between a receptor atom i and a ligand atom j is calculated from a pair-correlation (radial distribution) function, where ρ_{seg} is the number density of pairs ij that occur in a certain radial segment and ρ_{bulk} is the overall distribution when no interaction occurs. $f_{\mathrm{Vol\_corr}}$ is a correction factor for the ligand atom j, which was introduced in order to consider the volume occupied by the ligand itself. k_B is the Boltzmann constant and T is the absolute temperature. The final PMF score is calculated as a sum over all protein–ligand atom pairs ij within a certain cutoff radius.

Other knowledge-based potentials were introduced [70,81,102,133,200,213, 221, 306], mostly differing in the specific functional form of the atom pair potential or the size of the training dataset. Gohlke introduced the DrugScore [102] function with an additional nonpolar surface dependent single atom term to reflect the hydrophobic collapse. Muryshev [213] derived the potential function from pair-correlation studies, but retained some adjustable parameters which were then fitted to experimentally determined binding affinities.

2.5 Evaluation

There are three basic questions to consider when evaluating the reliability of a scoring function:

(i) Is the scoring function able to rate the docking pose most similar to the natural binding mode of a ligand with the best score?

(ii) Can the score of two ligands be used to decide which one binds better to a specific protein?

(iii) Is the score a direct measure for binding affinity such that results from different target proteins can be compared with each other?

As the amount of experimental data grows and becomes available to the scientific community in online databases [52,122,247,291,292,307], an increasing number of evaluation and comparison studies measuring the performance of different scoring functions have been carried out. Several studies on the topic of screening and docking evaluation have been published recently [29,82,146, 194,225,231,232,295,296].

It is a common conviction that there is no scoring function which performs well on all kinds of proteins. Each scoring function is able to account especially well for certain specific types of interaction patterns and protein classes. Furthermore, it is often found that combining the results of different scoring functions leads to more consistent results and the rate of false hits can be reduced by such consensus approaches. Often, a single scoring function is superior for the protein class of interest.

Therefore, a consensus score rarely performs as well as the best scoring function for a specific target, but might be the best choice if there is no additional knowledge about the target protein [51, 56, 273]. It is common sense that actual scoring functions are able to produce satisfying results, in particular, when they are trained to a specific problem. As scoring is such a major concern in computer-aided drug design, there is a great need for further improvements in this field.

3 Methods for Protein–Ligand Docking

The following section contains a survey of algorithms applied to the various types of docking problems as well as a short summary on scoring functions. The methods are typically related to specific software tools also mentioned here.

3.1 Rigid-body Docking Algorithms

Rigid-body docking algorithms have historically been the first approaches to screening sets of ligands with respect to their fit to a given target protein. The protein as well as the ligand is considered rigid, which reduces the problem to the search for the relative orientation of the two molecules with the lowest energy, involving six degrees of freedom.

Reviewing the development of more recent approaches to fragment docking, two enhancements were made. (i) More elaborate algorithms for searching in 3-D space, often adapted from other disciplines, were applied (see, e.g. Sections 3.1.2 and 3.1.3). These methods allow for a reasonable coverage of search space in short computing times. (ii) More information on the physicochemical properties of binding are included directly into the search process. Therefore, energetically unfavorable fragment placements are directly avoided, even if they make sense from a steric point of view. Both enhancements are essential for achieving a high performance for large variety of molecular fragments.

3.1.1 Approaches based on Clique Search

The docking problem can be understood as a problem of matching characteristic local features of the two molecules in 3-D space [165]. A match is an assignment of a ligand feature to a protein feature. Such a feature can either be a piece of volume of the active site of the protein or the ligand or an interaction between the molecules. The search procedure maximizes the number of matches under the constraint that they are compatible in 3-D space, i.e. that they can be realized simultaneously. In other words, compatibility means that a transformation can be found which simultaneously superimposes all ligand features to the matched protein features. In order to search for compatible matches, the following graph G is used: the vertices of $G$are all possible matches between the protein and the ligand; the edges connect pairs of vertices representing compatible matches. Mostly, compatibility means distance compatibility within a fixed tolerance ε: the matches (p_1, h_1),(p_2, h_2) are distance-compatible if and only if $|d(p_1, p_2) - d(l_1, l_2)| < \varepsilon$. A necessary condition for a set of matches to be simultaneously realizable is that all pairs of matches are distance-compatible. Therefore, an algorithm for enumerating cliques (fully connected subgraphs) can be applied to G. By superimposing the matched features of a clique one obtains an initial orientation of the ligand molecule in the active site of the protein.

One of the historically first and today probably most widely used software tool for molecular docking, the DOCK program [166], is based on the idea of searching distance-compatible matches. Starting with the molecular surface of the protein [61, 62, 246], a set of spheres is created inside the active site as shown in Figure 4. The spheres represent the volume which can be occupied by the ligand molecule. The ligand is either represented by spheres inside the ligand or directly by its atoms. In DOCK, an enumeration algorithm searches for sets with up to four distance-compatible matches. Each set is used for an initial fit of the ligand into the active site. Then the set is augmented by further compatible matches and the position of the ligand is optimized and scored. Since its first introduction, the DOCK software has been extended in several directions. The matching spheres can be labeled with chemical properties [267] and distance bins are used to speed up the search process [198, 266]. Recently, the search algorithm for distance-compatible matches changed to the clique-detection algorithm introduced by Kuhl and coworkers [78, 165].

An interesting algorithmic extension was introduced by Knegtel [155]. Instead of using a single protein, an ensemble of protein structures is used for

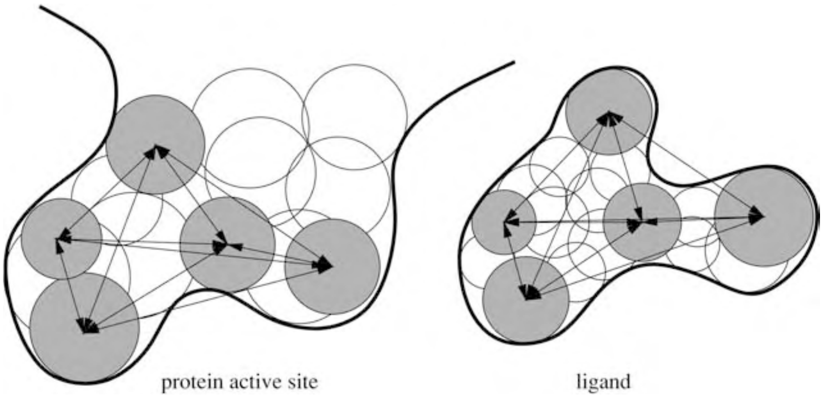


Figure 4 Sphere matching with the DOCK algorithm. (a) A protein active site is shown filled with spheres as they are used in the DOCK algorithm. (b) A ligand covered with spheres is shown. An example for a distance-compatible matching is highlighted in grey. All distances compared are shown by arrows.

the docking calculation. By averaging over the structures, a soft potential representing all structures at the same time can be constructed. With this approach limited protein flexibility can be taken into account. Two of the flexible ligand docking approaches based on DOCK will be introduced below. Furthermore, several scoring functions have been applied in combination with the DOCK algorithm [113, 199, 200, 266].

While the initial DOCK algorithm uses volume as the feature to be matched, other approaches use chemical interactions. Mizutani and coworkers [202] presented the program ADAM in which hydrogen bonding is the feature used for matching. Possible matchings are enumerated and filtered based on distance compatibility.

Further examples for distance-compatibility as an initial screen for fragment orientations are the well-known *de novo* ligand design program LUDI [33, 36] and the rigid-body docking program CLIX [170]. While LUDI's placement is based on matching hydrogen bond vectors and hydrophobic points, CLIX uses energetically favorable regions for functional groups of the ligand for its analysis. These regions are precalculated with the computer program GRID [106] employing a force field potential (see also Section 2). CLIX uses only two points for an initial matching. After fitting the two matched sites, the rotation about the common axis between the matched sites remains as an open degree of freedom. This rotation is then sampled in regular intervals.

3.1.2 Geometric Hashing

Geometric hashing [169] originated from computer vision and was first applied to molecular docking problems by Fischer and coworkers [84, 85]. In computer vision, the geometric hashing scheme was developed for the problem of recognizing (partially occluded) objects in camera scenes. For simplicity, we explain the geometric hashing algorithm for the 2-D case first and describe its application to the 3-D molecular docking afterwards.

Given a picture of a scene and a set of objects which can occur therein (called *models* in this context) both represented by points in 2-D space, the goal is to recognize some of the models in the scene. In a preprocessing phase, a hash table is created from the set of models. For each model, each pair of points defines a so-called *basis*. Then, for each basis, every third point belonging to the model is expressed in coordinates relative to the basis. A *tuple* (model, basis) is stored in a hash table addressed by the relative coordinates of the third point. The reason for having several bases for a model instead of a single one is that it is unknown in advance whether a part of the model is occluded in the scene.

In the recognition phase the scene is analyzed as follows. Every pair of points is considered as a basis. Once the basis is defined, all other points can be expressed by relative coordinates with respect to the basis resulting in a query for the hash table created before. The query votes for all matching tuples (model, basis) stored in the hash table. Finally, models with many votes are extracted, a transformation is calculated from the matching points, and the match is verified.

Two aspects make geometric hashing attractive for molecular docking problems: it is time-efficient and it deals with partial matchings standing for partially occluded objects in the terms of pattern recognition. The latter is extremely important because in most docking applications not all of the ligand features are matched with the protein since parts of the ligand surface are in contact with bulk water.

In order to apply geometric hashing to molecular docking, Fischer and coworkers [84, 85] used the sphere representation of DOCK as the underlying model. As docking is performed in 3-D space, three points (here spheres or atoms) are necessary to define a basis. As a consequence, the number of hash table entries increases with the fourth power of the number of ligand atoms, resulting in unacceptably large hash tables. Therefore, the basis is described by only two points leaving one degree of freedom open (rotation around the axis defined by the two points). With this model in mind, the geometric hashing approach can be directly applied to the molecular docking problem.

3.1.3 Pose Clustering

Pose clustering [181] is a different approach originating from pattern recognition that has been applied to the molecular docking problem [245]. Pose clustering was originally developed for detecting objects in pictures with an unknown camera location. The algorithm matches each triangle of object

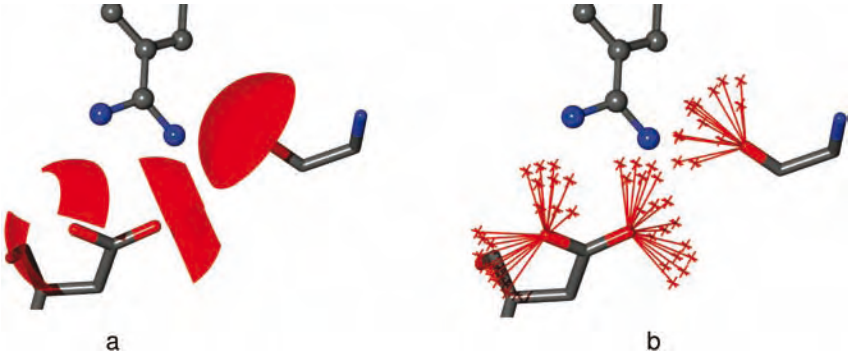


Figure 5 Interaction surfaces and points. Interaction surfaces of three hydrogen-bond acceptors in thrombin (shown in red). Protein atoms are drawn with sticks, ligand atoms with balls and sticks. (a) The interaction surfaces itself and (b) the approximation by interaction points used in the pose clustering algorithm.

points with each triangle of points from the picture. From a match, a camera location can be computed such that the triangles superimpose. The camera locations are stored and clustered. If a large cluster is found, the object is identified and the orientation of the camera with respect to the object is determined.

For applying pose clustering to molecular docking, the LUDI model of molecular interactions [33, 36] is used as the underlying representation. For each interacting group, an interaction center and an interaction surface is defined (see Figure 5). The interaction surfaces of the protein are approximated by discrete points, which then form the scene in the pose clustering algorithm. The centers of the ligand interactions are the object points which have to be matched to the scene.

While in the pattern recognition problem each triangle of object points can be matched to each triangle of picture points, in the docking application the matches are limited in various ways; (i) the interaction types must be compatible and (ii) matching triangle edges must have approximately the same length. A hashing scheme is necessary to efficiently access matching protein interaction surface triangles (picture points) for a given ligand interaction center triangle (object points). The hashing scheme stores edges between two points, and addresses them by the two interaction types of the points and the edge length. A list-merging algorithm then creates all triangles based on lists of fitting triangle edges, for two of the three edges of the query triangle. For a match, additional directionality constraints for the three interactions are checked. Then, a transformation is calculated that superimposes the two triangles. In the original application of pose clustering, the transformation is used to calculate the camera location. In molecular docking, the transformation directly describes the location of the fragment in the active site.

Finally, the transformations must be clustered. We use complete-linkage hierarchical clustering for this task. Up to a user-given distance-threshold, the two closest clusters are merged to a single one. The distance between two clusters is defined as the maximal distance between their objects. As a distance measure between transformations, the root-mean-square deviation (RMSD) of the atoms after applying the transformations is used. After a linear-time preprocessing phase, this quantity can be calculated from the transformations in constant time [245]. For each of the clusters generated, the typical postprocessing steps are performed, like searching for additional interactions, checking for protein–ligand overlap and scoring.

3.1.4 **Fast Shape Comparison**

In molecular similarity, fast methods for comparing molecule shapes are of great interest. A strategy developed in the 1980s is to model atoms by Gaussian functions instead of van der Waals spheres [144, 154]. This method leads to volume-based similarity measures like the Carbo and Hodgkin Indices [49, 126], and allows for fast numerical optimization.

The concept of Gaussian shapes can be adapted to molecular docking [195]. The scoring function measuring the quality of fit consists of two terms. The first quantifies the volume overlap between the protein and the ligand which has to be minimized. The second measures an area intersection which has to be maximized in order to achieve a tight fit. The overall function is differentiable and can therefore be used in combination with fast numerical optimization. Gaussian shapes are the underlying concept of the fast rigid-body docking software FRED which is frequently used [1].

3.1.5 Superposition of Point Sets

In each of the discussed algorithms for rigid-body docking, the superposition of point sets is a fundamental subproblem that has to be solved efficiently and therefore is discussed here briefly.

The superposition problem can be described as follows: Given two sets X,Y with n vectors each, find a transformation $T = \Omega$,t minimizing the root-mean-square deviation between $X$and the transformed vector Y: $RMSD_{X,Y}(T) = \sqrt{(1/n)\sum_i(x_i - \Omega y_i - t)^2}$. Let Ω describe a rotation around the centroid of Y, then $t' = (1/n) * (\sum_i x_i - \sum_i y_i)$ minimizes $RMSD_{X,Y}(T)$ for all rotations Ω.

Optimizing Ω is a more difficult task. Ferro and Hermans [83] and later Sippl and Stegebuchner [269] proposed iterative algorithms rotating subsequently around the x-, y- and z-axes. If the axis is fixed, the optimal rotation angle can be determined analytically.

Kabsch [141, 142] formulated the problem as a constrained optimization problem. Using Lagrange multipliers and the calculation of eigenvectors, Kabsch was able to solve the problem directly. Kabsch's algorithm is very time-efficient and used in several codes for molecular docking today. A recent further development based on Kabsch's ideas combined with quaternions for describing rotations can be found in Ref. [65].

3.2 Flexible Ligand-docking Algorithms

The major limitation of rigid-body docking algorithms is that the conformational flexibility of the ligand molecule is not considered. Often small molecules have large conformational spaces with several low-energy states. Significant differences between the bound conformation and the calculated low energy conformation can occur even for small molecules with only a few rotatable bonds [286].

3.2.1 Conformation Ensembles

In principle, ligand conformational flexibility can be incorporated by applying rigid-body docking algorithms to ensembles of rigid structures, each representing a different conformation of the same ligand. The size of the ensemble is critical, since the computing time increases linearly with the number of conformations, and the quality of the result drops with increasing difference between the most similar conformation of the ensemble and the complex conformation.

Miller and Kearsley developed the Flexibase/FLOG docking algorithm based on conformation ensembles. Flexibases [145] store a small set of diverse conformations for each molecule from a given database. The conformations are created with distance geometry methods [66, 119, 120] which will be introduced later in the context of docking. Then up to 25 conformations per molecule are selected by RMSD dissimilarity criteria. Each conformation of a molecule is then docked using the rigid-body docking tool FLOG [203] which is similar to the DOCK algorithm discussed above.

A different approach based on conformation ensembles was presented by Lorber and Shoichet [184]. Here, about 300 conformations per molecule are created on average for a database of molecules. For each molecule, a rigid part, e.g. an aromatic ring system, is defined. The conformation ensemble is created such that the atoms of this rigid part are superimposed.

Then, the DOCK algorithm is applied to the rigid part and all conformations are subsequently tested for overlap and finally scored. With this method, a significant speedup can be achieved compared to an independent docking of the conformations.

An important point in this scheme is the dependence between the conformation generation algorithm and the docking algorithm: the fewer conformations are created, the higher the tolerance in the matching phase of the docking algorithm must be. Therefore, the algorithms applied to the two subproblems are related making it necessary to tune parameters describing the coverage of conformational space in accord with the tolerance in protein–ligand overlap.

A key element for the success of these methods is obviously the conformation generator. Frequently used are systematic approaches like ROTATE [3] and OMEGA [2]. As a rule of thumb, a few hundred conformations have to be generated in order to have a representative below 1.5 Å RMSD compared to the bound conformation.

The probably most frequently used docking tools based on conformational ensembles are GLIDE [87,114] and FRED [1,195]. GLIDE uses conformational ensembles for fast prescreening of ligand poses. The most promising candidate poses are then numerically optimized and scored. Further examples of software tools following the ensemble strategy are EUDOC [222] and the method by Diller [72].

3.2.2 **Flexible Docking based on Fragmentation**

A possible way to handle conformational flexibility directly in the docking algorithm is by fragmentation. Here, the ligand is divided into several fragments. Each fragment is either rigid or has only a small number of conformations which can be handled by a conformation ensemble. Obviously, the fragment-based docking approaches and *de novo* ligand design are closely related. The major difference is that, while in the docking algorithm the fragments stem from a single molecule, the *de novo* ligand design algorithm picks a fragment out of a database. *De novo* design algorithms will be discussed in Section 5.2.

3.2.2.1 **"Place & Join" Algorithms**

With respect to the way in which the fragments are reconnected during the docking calculation, we can distinguish between two classes of algorithms. In a "place & join" algorithm, each fragment is docked independently. Then, placements for adjacent fragments in which the connecting atoms overlap are identified and reconnected.

The first algorithm of this kind was developed by DesJarlais and coworkers [69]. The ligand is manually divided into two fragments having one atom in common. Then, placement lists are created for each fragment using the docking algorithm DOCK. The algorithm searches through these lists for placement pairs in which the common atom is located approximately at the same point. Finally, the fragments are reconnected, energy minimized, and scored.

Sandak and coworkers [254–256] applied the geometric hashing paradigm to develop a "place & join" algorithm. As before, the ligand is divided into fragments with one overlapping atom, called the *hinge*. For each ligand atom triplet of a fragment, a hash table entry is created and addressed with the pairwise distances between the atoms. The entry contains a fragment identification as well as the location of the hinge. In the matching phase, protein sphere triplets are used to extract ligand atom triplets with similar distances. As a result a vote is counted for a hinge location for each match. Hinge locations with many votes are then selected, and the fragments are reconnected accordingly and finally scored.

"Place & join" algorithms are advantageous in cases in which the molecule consists of a small set of medium-sized rigid fragments. If the fragments are too small, it is difficult to place them independently. Another difficulty is to generate correct bond lengths and angles at the connecting atom without destroying the previously found interactions of the fragments to the protein.

3.2.2.2 **Incremental Construction Algorithms**

The second kind of fragment-based docking algorithms – the one which is used most frequently today – follows the incremental construction method. Instead of placing all fragments of the ligand independently, the incremental construction algorithm starts with placing one fragment (called *base* or *anchor fragment*) into the active site of the protein. Then, the algorithm adds the remaining parts of the ligand to the already placed fragment iteratively. Thus, an incremental construction algorithm has three phases: the selection of base fragments, the placement of base fragments and the incremental construction phase. An incremental construction algorithm can start with several base fragments; however, in contrast to the "place & join" algorithms the placements are not combined, but taken as anchoring orientations to which the remaining parts of the ligand can be added.

Incremental construction originated from the area of *de novo* ligand design. Moon and Howe [204] presented the peptide design tool GROW based on this strategy. The first docking algorithm based on incremental construction was developed by Leach and Kuntz based on the DOCK program [174]. First, a single anchor fragment is selected manually and docked into the active site using a variant of the DOCK algorithm which handles hydrogen-bonding features in the matching phase. A subset of placements is selected for which the incremental construction phase is started. For this phase, a backtracking algorithm is used that enumerates the space of nonoverlapping placements of the whole ligand in the active site. After adding a fragment to the current placement, a refinement routine is used to eliminate steric strain and to improve hydrogen bond geometries. The final placements are then filtered, refined and scored with a force field-based approach. Although there are several manual steps in this procedure, the work demonstrated that the incremental construction idea can be applied to the docking problem.

Leach published a second docking algorithm [172] similar to incremental construction with respect to the way that the degrees of freedom are fixed sequentially. Degrees of freedom considered are the ligand orientation and conformation described by a discrete set as well as a set of rotamers for selected protein side-chains. Leach used a variant of a "branch & bound" scheme, called A* algorithm with dead-end elimination, to search efficiently through the space of possible configurations.

The docking algorithm contained in the FlexX package [**?**, 239, 241] is also based on incremental construction. FlexX is a fully automated approach to molecular docking developed for virtual screening purposes. In the first phase, a small number of base fragments are selected. An efficiently computable scoring function is used to select fragments which are suitable for placement. Base fragments should contain a reasonably large number of interacting groups and at the same time a relatively small number of low-energy conformations. A necessary condition for a successful calculation is that the selected base fragment binds to the protein and is not mostly exposed to water in the final protein–ligand complex. In order to ensure this, a small set of base fragments distributed over the ligand is selected.

For placing the base fragments, the pose clustering algorithm is applied. Base fragment conformations are enumerated within the placement algorithm. The advantage of the pose clustering algorithm is that it is based on the molecular interactions instead of the shape of the fragment. This facilitates the handling of much smaller fragments than in shape-based algorithms, down to the size of a single functional group. All calculated placements up to a given number form the input to the incremental construction phase. In contrast to Leach's algorithm, a greedy strategy is applied always selecting the k placements with the highest estimated score ($k \sim 800$). Each iteration of the incremental construction algorithm contains the following steps: adding the next fragment in all possible conformations to all placements from the previous iteration (or the base placement phase), searching for new protein–ligand interactions, optimizing the ligand position to improve the interaction geometries and reduce steric strain, selecting a subset of placements with high score, and clustering the placements to achieve a reasonable degree of diversity in the solution set. The overall search strategy is shown in Figure 6.

Ligand conformations within FlexX are based on the MIMUMBA model [153]: To each rotatable bond, a set of low-energy torsion angles is assigned, previously derived from a statistical analysis of the Cambridge Structure Database (CSD) [7]. Ring system conformations are precomputed using the 3-D structure generator Corina [89, 252]. For scoring protein–ligand complexes, a variant of Böhms empirical scoring function [34] is used.

Several extensions of the FlexX approach have been developed. The interaction model has been extended such that hydrophobic fragments can be

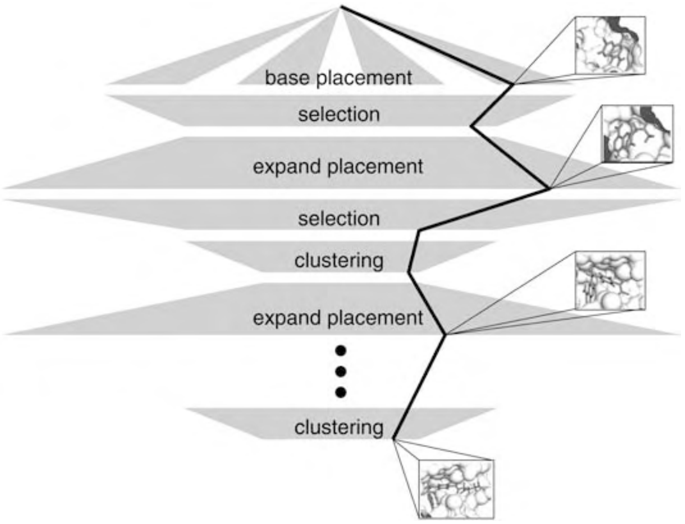


Figure 6 Incremental construction. Flexible ligand docking by incremental construction. The search tree resulting after the steps of the algorithm as described in the text are shown in grey. The black line illustrates the construction process for one placement of the complete ligand.

handled with the pose clustering algorithm [238]. Further development of the interaction scheme allows for even faster virtual screening. FlexX-Scan is a special FlexX variant which makes use of a new interaction scheme resulting in speedups of up to a factor 4 [257]. In order to place critical water molecules and metal ions located in the protein–ligand interface, the "particle concept" has been invented and integrated into the FlexX software. The particle concept allows for automatic placement and energetic consideration of approximately spherical objects during the docking calculation [240]. FlexE is an extension of FlexX regarding the handling of protein flexibility [57, 58]. Similar to Knegtels approach mentioned above, FlexE takes an ensemble of slightly differing protein structures as input. However, the structures are not numerically merged to a single description. FlexE uses various graph algorithms to explicitly consider the different alternative conformations for protein parts like side-chains or small loop fragments and combine them to a single protein conformation which is best suited for the protein–ligand complex created during the docking calculation.

Two other approaches based on incremental construction have been published. The program Hammerhead [300] and its successor Surflex [135] differ from FlexX in the construction strategy. Instead of adding small fragments (cut between each rotatable bond), the ligand is divided into a small set of large fragments. During the construction phase, the next fragment is added such that the connecting atom (or bond) overlaps and interactions to the protein can be formed. Therefore, there is no discrete sampling at the torsion angle of the added fragment. However, since the bond angle at the overlapping atom may vary, high-energy conformations will also be generated and the situation in which a fragment does not interact directly with the protein is more difficult to handle.

Makino and Kuntz [191] and later Ewing and Kuntz [79] presented a fully automated incremental construction docking algorithm based on backtracking. A single anchor fragment is selected maximizing hydrogen bonding features. During the incremental construction phase, the number of conformations for each fragment is limited to reduce the size of the search space. This method is called a limited backtrack search. For scoring, the AMBER force field [224] is used with a modification allowing the handling of multiple protonation states.

Incremental construction algorithms are the basis for widely used docking tools like DOCK and FlexX. The quality of the predicted structures strongly depends on the number of different placements considered in each iteration of the incremental construction process. Therefore, although the overall concept of incremental construction is simple, much effort must be put in the time-efficient analysis of partially placed ligands, e.g. the protein–ligand overlap test, the evaluation of the scoring function and the postoptimization of the ligand placement.

In practice, incremental construction has proved to be a reasonable compromise between accuracy and computing speed. This holds especially for virtual screening applications, in which computing speed is of central importance. For accurate docking, the question about the loss in quality due to the greedy strategy during incremental construction arises. For FlexX, an implementation of an alternative "branch & bound" method shows that for several complexes no significant improvement can be achieved by a full enumeration of the solution space [111].

3.2.3 Genetic Algorithms and Evolutionary Programming

Since the mid-1990s, genetic algorithms have been applied to the molecular docking problem in several approaches [50, 55, 92, 137, 138, 205, 219, 280, 303]. A genetic algorithm [104] is a general-purpose optimization scheme which mimics the process of evolution. The individuals are configurations in the search space. A so-called fitness function is used to decide which individuals survive and produce offspring.

Several elements must be modeled in order to use the idea of genetic algorithms for an application like molecular docking. First of all, a linear description of a configuration (the *chromosome*) is needed describing all degrees of freedom of the problem. Finding the chromosome description is the most difficult modeling part. A suitable description is free of redundancy and models constraints of the configuration space directly such that configurations violating constraints are never generated during the optimization.

Second, a fitness function has to be developed. The fitness function is closely related to scoring functions for molecular docking with one extension. Scoring functions normally work on 3-D coordinates. Therefore, the chromosome of an individual has to be interpreted in order to apply the scoring function. This step is called the *genotype-to-phenotype conversion*. Since most of the computing time is spent on evaluating the fitness function, the conversion and the evaluation have to be done efficiently.

The optimization scheme itself is more or less independent from the application. Typically, several parameters have to be chosen like the population size, the number of generations, crossover and mutation rates, etc. Here, it is important to achieve a reasonable trade-off between optimizing the fitness function and keeping the diversity in the population.

The genetic algorithm which is probably most widely applied for molecular docking today was developed by Jones and coworkers [137, 138] and is implemented in the software tool GOLD. A configuration in GOLD is represented by two strings. The first string stores the conformation of the ligand and selected protein side-chains by defining the torsion angle of each rotatable bond. The second one stores a mapping between hydrogen-bond partners in the protein and the ligand. For fitness evaluation, a 3-D structure is created from the chromosome representation by first generating the ligand conformation. According to the mapping stored in the second string, hydrogen bond atoms are superimposed onto hydrogen-bond site points in the active site. Finally, a scoring function evaluating hydrogen bonds, the ligand internal energy as well as the protein–ligand van der Waals energy is applied as the fitness function. Recently, the Chemscore scoring function was implemented in GOLD, showing an improvement in the virtual screening performance [285].

Oshiro and coworkers [219] developed two variants of a docking method, both based on genetic algorithms and the DOCK approach. The variants differ in the way the relative orientation of the ligand to the protein is described. The first variant is similar to the GOLD algorithm and encodes the matching of ligand atoms to protein spheres in the chromosome. A superposition is used to generate the 3-D orientation of the ligand. The second variant stores the relative orientation directly by a translation vector and three Euler angles. For scoring, a simplified version of the AMBER force field was used.

Gehlhaar and coworkers [92] developed a docking algorithm based on evolutionary programming called EPDOCK. In contrast to a genetic algorithm, offspring are created from one parent by mutation only. Each member of a population is competing for survival in a so-called tournament. EPDOCK contains a self-developed scoring function based on atomic pairwise linear potentials for steric interactions and hydrogen bonding.

3.2.4 Distance Geometry

Distance geometry [119,120] is a well-known technique from the area of structure determination via NMR technology. Instead of describing a molecule by coordinates in Euclidean space, it is described by a so-called *distance matrix* containing all interatomic distances. Based on distance matrices, a set of conformations can be described in a comprehensive form by calculating a distance interval for each atom pair.

The distance geometry methodology can be directly used in the docking algorithm based on clique search and distance compatibility (see Section 3.1.1). Two matches between protein site points and ligand atoms are compatible if the site point distance lies within the distance interval of the ligand atom pair. The drawback of this approach is that the distance matrix with n^2 distances compared to $3n$ atom coordinates is overdetermined: fixing the atom–atom distance between a given atom pair to a single value causes other distance intervals to shrink. Since the exact new interval boundaries are difficult to calculate, the triangle and tetrangle inequality are used to approximate them [74,75]. In other words, only a very limited number of distance matrices can be converted back to 3-D space.

Ghose and Crippen [95] first worked on this approach on a more theoretical basis. Later, Smellie and coworkers [270] applied this methodology to real test cases. Billeter and coworkers [28] combined the description of molecules by distance constraints with an efficient algorithm for constrained optimization.

The screening software Specitope, later versions named SLIDE, developed by Schnecke and coworkers [259] uses distance matrix comparisons as a first filter step. However, the flexibility of molecules is not modeled by distance intervals. Instead, a weighting scheme is defined to scale down the contributions of more flexible atom pairs in the overall score.

3.2.5 Random Search

Once a scoring function for evaluating protein–ligand complexes is available, random search algorithms can be applied to the docking problem. Random placements can be either created directly by randomly fixing all degrees of freedom or they can be derived from a (random) starting structure by random moves. In most cases the structure generation is combined with a numerical optimization driving the placements to the closest local minima. Most approaches of this kind are MC algorithms discussed separately in Section 3.3.3.

Sobolev and coworkers [271] presented a random-search algorithm in their docking program LIGIN. A large set of starting structures is created randomly and then optimized in two steps. The surface complementarity is optimized first, the hydrogen bond geometry in a second step. So far, LIGIN does not include ligand flexibility although the method is in principle able to handle it.

In order to avoid the repeated evaluation of very similar structures, Baxter and coworkers [22] use a method called *tabu search* in their docking software called PRO_LEADS. Starting with an initial random structure, new structures are created by random moves. In tabu search optimization, a list (the tabu list) is maintained containing the best and most recently visited configurations. Moves which result in configurations close to one in the tabu list are rejected except if they are better than the best scoring one. The tabu list technique improves the sampling properties of the random search algorithm in that it avoids revisiting configurations.

3.3 Docking by Simulation

While all previous methods for the docking problem are based on some kind of combinatorial optimization algorithm, there are several approaches tackling the problem by simulation techniques. Instead of trying to enumerate a discrete low-energy subspace of the problem, these approaches begin their calculation with a starting configuration and locally move to configurations with lower energy.

3.3.1 Simulated Annealing

Simulated annealing [150] is a well-known simulation technique which is also frequently used for solving complex optimization problems without any physical interpretation of the simulation itself. The overall simulation routine iterates the following steps. Starting with a configuration A with an energy or score value $E(A)$, a random local move to a new configuration B with energy $E(B)$is calculated. The acceptance of the new configuration is based on the Metropolis criterion, which means that the configuration is accepted if $E(B) \leq E(A)$ or with probability $P = \mathrm{e}^{-(E(B) - E(A))/(kBT)}$ otherwise where k_B is the Boltzmann constant. Over simulation time, the temperature T is reduced based on a so-called cooling schedule such that accepting configurations with increased energy becomes less likely.

The AUTODOCK program for protein–ligand docking developed by Goodsell and coworkers [107,108,206] is based on this strategy. For energy calculation, molecular affinity potentials [106] are precalculated on a grid. Yue developed a program for optimizing distance constraints for rigid-body docking based on simulated annealing [304]. There are several methodical improvements of simulated annealing developed over time. One example which was also used for molecular docking is stochastic tunneling [192,201,202] which rescales the potential energy surface in order to improve the probability of transition between local minima.

3.3.2 MD Simulations

In principle, molecular docking problems can be solved with MD simulations. In fact, the earliest approaches for predicting protein–ligand interactions with the computer were based on MD calculations [234].

In MD, a force field is used to calculate the forces on each atom of the simulated system. Then, following Newtonian mechanics, velocities and accelerations are calculated, and the atoms are moved slightly with respect to a given time step. Introducing MD and force fields is clearly beyond the scope of this chapter. However, some aspects of docking by MD simulations will be mentioned briefly. The simulation becomes more exact, the smaller the time step and the more atoms of the system are taken into account. Thus, MD simulations can become very time consuming and are therefore not appropriate for inspecting large sets of molecules.

In order to avoid very long computing times, methods performing greater moves of the ligand in a single step have been developed. This decreases the dependence of the outcome of the calculation from the starting structure and allows a better sampling of the search space. Di Nola and coworkers [71, 193] developed a technique for this purpose called "helicopter view". For a limited time, the temperature of the system is increased for selected degrees of freedom (protein–ligand relative orientation) and the repulsive terms of the energy function are decreased. This enables the algorithm to escape from local minima in the energy function. A similar effect can be achieved by shrinking and growing the ligand inside the active site of the protein [220].

Given and Gilson have developed a four-phase docking protocol based on MD [100]. First, a set of low-energy ligand conformations is created using MD with alternated heating (in order to perturb the structure) and cooling (in order to minimize the structure). Then, the ligand is placed randomly into the active site and several times minimized. In the final phase, the most stable configurations are investigated further using MD with alternate heating and cooling. The goal of the last phase is to explore the search space around the stable conformations in more detail.

A frequently used technique of speeding up MD simulations is the precalculation of force-field contributions from protein atoms on grids. The force acting upon a ligand atom can then be efficiently calculated by a simple table-lookup instead of summing over all protein atom contributions. The potentials, however, can only be precalculated for atoms which do not change their orientation in space. In order to avoid complete neglect protein flexibility, Luty and coworkers [186,298] divided the protein into a rigid and a flexible part. Every atom sufficiently far away from the active site is considered as fixed in space; all force-field contributions of these atoms can be precalculated on a grid. During the MD simulation, only the active site atoms of the protein and the ligand atoms are allowed to move and have to be considered explicitly in the force-field calculation.

3.3.3 MC Algorithms

In an MD simulation, the local movements of the atoms are performed due to the occurring forces. In contrast, in a MC simulation, the local moves of the atoms are performed randomly. The simulated annealing algorithm discussed above is one special variant of an MC simulation. Two components are of major importance in the development of an MC algorithm: the description of the degrees of freedom and the energy evaluation.

Concerning the degrees of freedom, the aim is a method that avoids sampling high-energy states. A good example of how to realize this concept is the description of the conformational space of a molecule by internal coordinates (bond lengths, bond angles and torsion angles) shown in Figure 7 instead of by Cartesian coordinates for all atoms. With internal coordinates, each type of variable is related to a different energy scale. Changing a bond length is energetically more expensive than changing a bond angle, which in turn is energetically more expensive than changing a torsion angle. Using internal coordinates, it is possible to navigate through the low-energy conformational space by defining the amounts by which the variables of each type are changed. Thus, internal coordinates are the more appropriate description form for MC algorithms.

There are several examples for MC-based docking algorithms. Hart and Read [118] developed an MC scheme combined with simulated annealing. In the MC run, random orientations are created and moved such that protein–ligand overlap is reduced. The results are then optimized in a simulated

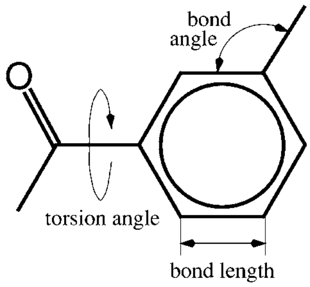


Figure 7 Internal coordinates. Internal coordinate representation of a molecule. The 3-D structure can be constructed from bond lengths, bond angles and torsion angles. Most of the computing time during an MC simulation is spent in the calculation of the energy (or score) of a state. Therefore, this step has to be as time-efficient as possible. Often, energy potentials are precalculated on a grid to speed up this step.

annealing optimization scheme. Protein–ligand overlap and scores are calculated based on precalculated grids. McMartins and Bohaceks QXP program [196] implements a similar approach which can be applied to molecular docking and structural superposition. Wallqvist and Covell [289] also use MC for optimizing the final ligand orientation. Instead of a random set of starting structures, a surface matching algorithm is used.

Abagyan and coworkers [5] developed the software package ICM combining the MC algorithm with an internal coordinate description such as explained above. In contrast to other approaches making purely random moves, ICM is able to make moves based on probability distributions for variable sets. For example, a probability distribution for a torsion angle can be defined such that low-energy torsion angles are more likely to occur than high-energy torsion angles. This biases the MC calculation towards low-energy states.

MC algorithms can be used to overcome the limitation of MD simulations to get stuck in local minima. In order to get low-energy structures, MC can then be combined with energy minimization as described by Apostolakis and coworkers [11]. After a random creation of starting structures, a minimization with a modified van der Waals potential is performed. The energy function for the evaluation of the conformations is a sum of three terms: a force-field energy, a hydrophobic solvation term which is proportional to the SAS of the complex and an electrostatic solvation term obtained from the solution of the linearized Poisson–Boltzmann equation. The best scoring configurations are further analyzed with a MC simulation method called MCM. MCM performs MC interleaved with minimization steps and uses the energy function mentioned above.

PRODOCK is an MC-based docking algorithm developed by Trosset and Scheraga [283, 284]. The software uses either the AMBER IV or ECEPP/3 force field with a grid-based energy evaluation. For calculating energy values within the grid, Bezier splines were used allowing for a more accurate estimation of the energy value as well as information about the derivatives at this point. Like ICM, an internal description of the degrees of freedom is used. The MC simulation is interleaved with energy minimization steps like in Apostolakis' approach.

Brutlag and coworkers proposed a stochastic roadmap simulation (SRS) as an efficient tool for studying molecular motion [9, 10]. A directed graph (*roadmap*) is calculated, where each node represents a randomly sampled conformation of the ligand and the associated protein using internal coordinates. The edges are labeled with transition probabilities, which correspond to the relative motions between such conformations. The probabilities are calculated between every pair of neighboring nodes using Boltzmann statistics. As the roadmap implicitly defines a Markov chain, steady-state occupancies can be calculated without simulating thousands of possible paths through various conformational states.

3.3.4 Hybrid Methods

Due to the complexity of the docking problem, all methods have their pros and cons. Fragment-based approaches and genetic algorithms achieve a wide coverage of the configuration space; however, simulation-based methods outperform others in finding exact local minima, i.e. predicting an exact placement. Combining different methods is therefore a reasonable approach which can result in methods containing the best of each.

Two approaches have been published recently which combine rapid fragment-based searching techniques with sophisticated MD or MC simulations.

Wang and coworkers [290] developed a multistep approach based on rigid-body docking and MD. First, a set of low-energy conformers is created. Each conformer is docked rigid into the active site using DOCK. The high-scoring placements are then optimized using an MD-based simulated annealing optimization in combination with the AMBER force field.

Hoffmann and coworkers [127] combined the incremental construction algorithm in FlexX with an MD-based procedure for postoptimization. First, FlexX is used to create a sample of a few hundred ligand placements. The goal of the second phase is to improve the overall ranking of the solutions and to identify the correct placement. The placements are first energy-minimized using the CHARMM force field. For re-ranking, the software package CAMLab [128] is used. The final score consists of three contributions: a force-field energy, an electrostatic part and a nonpolar part of the solvation energy. The electrostatic part of the solvation energy is calculated by solving the Poisson equation using fast multigrid methods, while the nonpolar part is approximated by the total solvent-accessible surface.

4 Structure-based Virtual Screening

A molecular docking algorithm and a scoring function are obviously the key elements in a structure-based virtual screening software package. In order to make it complete, technical issues like parallel computing on large compute clusters and data management have to be resolved. On a second look, one also identifies slightly different algorithmic problems worthwhile to address: considering pharmacophoric constraints and the exploitation of compound similarity for accelerating the screening process.

4.1 Considering Pharmacophoric Constraints

A frequent scenario for structure-based virtual screening is that, apart from the protein structure, key interacting groups within the active site are known. We would like to guide the docking algorithm to only create poses in which the ligand forms interactions to these key groups. This will focus the resulting compounds to those able to form the interactions requested by the modeler. Since these additional pharmacophoric constraints reduce the number of degrees of freedom, they should also reduce the computational demands as a positive side-effect.

In an early version of DOCK, there was the option to mark some of the protein active site spheres as essential (so-called *red spheres*). The clique detection algorithm can limit its search to those cliques containing the red spheres. If the red spheres are in the clique, they are matched and therefore the required interaction is formed. In addition, less computing time is required for the clique detection.

In a flexible docking algorithm, the situation is more complicated. Obviously, the pharmacophoric constraints can be checked after the pose was generated. This is inefficient, however, since the constraints are applied at a very late phase of the algorithm. The first algorithmic approach for considering pharmacophoric constraints in a flexible docking algorithm was FlexX-Pharm [123, 124]. FlexX-Pharm allows for a variety of pharmacophoric constraints (hydrogen bonding, hydrophobic and steric spots, chemical group locations) combined in a logical expression. During the incremental construction algorithm, for each partially built-up ligand, FlexX-Pharm keeps book about the constraints already obeyed and those which still have to be obeyed in order to achieve a valid solution. FlexX-pharm performs a cascade of fast logical and geometric checks in order to rule out the partial solution which cannot be extended to a valid solution. Today, most molecular docking tools allow for the consideration of pharmacophoric constraints.

4.2 Docking of Combinatorial Libraries

The development of combinatorial chemistry and its application to drug design [88, 109] has led to new search problems in the context of molecular docking. An example of a combinatorial library is given in Figure 8. The number of molecules which can be synthesized on the combinatorial chemistry bases has increased dramatically compared to classical methods. Therefore, any screening methodology has to face many more molecules.

Probably most important for the development of docking methods is the introduction of formal structure into this increased search space. If an unstructured compound collection is given, each molecule has to be analyzed

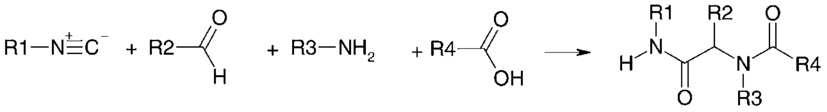


Figure 8 Combinatorial libraries. Based on the Ugi reaction, combinatorial libraries with four different R-groups can be created. After the reaction, all library molecules have the core shown on the right in common, but differ in the four R-groups attached to it.

independently in a screening experiment. Combinatorial libraries, however, follow a systematic build-up law for synthesizing molecules from a limited set of building blocks. This structure can be exploited to drastically reduce the runtime of virtual screening calculations.

In the context of combinatorial libraries, one can distinguish between three kinds of docking problems:

- *Combinatorial docking problem*: given a library, calculate the docking score (and the geometry of the complex) for each molecule of the library.
- *R-group selection problem*: given a library, select molecules for the individual R-groups in order to form a smaller sublibrary with an enriched number of hits for the target protein.
- *De novo library design problem*: given a catalog of molecules, design a library (including the rules of synthesis) optimizing the number of hits for the target protein.

Methods for these problems have emerged from the area of molecular docking and *de novo* ligand design (see Ref. [163] for an early overview on combinatorial docking methods). In the first case, the docking algorithms are applied to individual molecule fragments like R-groups or the core of the library and the resulting information is then combined yielding placements for individual library molecules. The methods for combining the placements differ. As in the case of docking algorithms, they can also be classified as either a "place & join" or an incremental construction method. In the latter case, the *de novo* ligand design is constrained by predefined rules of synthesis.

Early algorithms for the combinatorial docking problem analyzed the similarity in given ligand datasets in order to speed up the search process. The focus in these papers is on structurally relating ligands within the dataset. One approach to do so is to generate a minimal tree structure representing the whole ligand dataset [237]. Another approach is to speed-up conformational searching based on clustering similar molecules [190]. In both cases, the derived hierarchy of molecules can then be used in an incremental construction docking method.

The combinatorial docking tools PRO_SELECT [211] and CombiDOCK [278] are based on the incremental construction method. In both approaches, a library is formed by a template (or core) molecule with a set of attachment points to which one out of a predefined set of substituents can be connected. The template is then positioned inside the active site without considering its substituents. Starting from a few orientations of the template, the substituents are placed into the active site of the protein independently. In case of PRO_SELECT, substituents are then selected based on score and additional criteria like 2-D similarity and feasibility of synthesis. CombiDOCK calculates a final score for whole library molecules by combining fragment scores.

An algorithm that [244] is part of a combinatorial docking extension of FlexX, called FlexX-C, handles the library as a rooted tree in its closed form. Molecules from the library required during the docking calculations are created on the fly. With this method libraries with a few hundred thousand molecules can be handled in main memory during a docking calculation. The tree representation allows for handling complex libraries consisting of several R-groups which can be arbitrarily linked with the one limitation that no ring closures over different R-groups are allowed. The recursive combinatorial library algorithm is a natural extension of the incremental construction method to multiple molecules – in each incremental construction step, all possible R-group molecules are added sequentially [244].

Several approaches based on ligand *de novo* design software have been published for R-group selection problems. Kick and coworkers [149] applied a variant of the BUILDER program [248] to the preselection of substituents for a library targeted to Cathepsin D. Böhm [41] applied the LUDI program to the docking of two groups of fragments which can be connected pairwise in a single-step reaction to search for new thrombin inhibitors. In principle, all programs for fragment-based *de novo* ligand design can be applied in a similar way to the R-group selection problem.

Finally, we mention two methods for *de novo* library design. Caflisch [48] applied the MCSS technique generating fragment placements which are subsequently connected. The DREAM++ software [190] combines tools for fragment placement and selection. The selection process is done such that only a small set of well-characterized organic reactions are needed to create the library.

4.3 Database Approaches

When a large compound library has to be screened, the question arises whether commonalities of compounds can be used, in principle, to speed up the screening process. Like in the case of combinatorial libraries, we would like to perform computations on similar or identical parts of compounds only once in order to improve the efficiency of the overall calculation.

For some docking approaches, the algorithmic idea behind them nicely extends to the case of virtual screening. The probably best example for such a case is geometric hashing (see Section 3.1.2). In this method, multiple compounds can be added to the geometric hash table and will be retrieved by the appropriate protein-based query. The program SLIDE [259] basically follows this strategy for handling multiple compounds.

For fragment-based docking algorithms, a possible strategy would be to avoid the recalculation of poses for common fragments. Lorber and Shoichet demonstrated this approach for a multi-conformation docking tool NWDOCK [184, 277] (see also Section 3.2.1). They extracted rigid fragments from molecules and docked the fragments individually. All molecules containing the fragment are then superimposed on the fragment pose reusing the placement information calculated once for the fragment. With FlexX, we found out that the storage and retrieval of base fragment poses can reduce the required computing time by a factor of 3–4 depending on the library used [258].

Alternatively to looking at common fragments, one can also look for common pharmacophores. The software PHDOCK by Joseph-McCarthy and coworkers [140] follows this idea. A multi-conformer database of compounds is organized by common pharmacophores. The docking algorithm (in this case the DOCK method) can first match pharmacophores. In case of a matching pharmacophore, the compounds containing it are retrieved and further examined. This small selection of methods already shows that understanding virtual screening as an integrated problem is a promising research direction towards even faster methods.

5 From Molecules to Fragment Spaces: Structure-based *De Novo* Design

Instead of moving from single-molecule docking to virtual screening and database approaches to cover a larger part of the chemical universe, a complimentary strategy can be the *de novo* assembly of compounds within an active site of a protein. This has already been mentioned several times throughout this chapter and will be addressed in the following.

A recent review on *de novo* design methods can be found in Ref. [260]. Here, we will focus on summarizing the concept of fragment spaces and algorithms to search them. In addition, the issue of synthetic accessibility will be surveyed and a couple of application scenarios will be described briefly.

5.1 Modeling Fragment Spaces

The difference between a compound library and a fragment space is rather small, but significant. A compound library is basically a collection of molecules, whereas a fragment space contains a collection of fragments. In contrast to the definition given earlier, here, a more general definition of the term fragment is useful. Concerning size, a fragment can be rather large or it can be a single atom only. In addition, a fragment in this context has one or several attachment points, each of a certain type. In some cases, these attachment points are just indicated by missing (hydrogen) atoms and the corresponding chemical environment, therefore, defines the type. In others cases, additional link atoms, sometimes also called *links*, are used which are essentially dummy atoms of certain types connected to the attachment points of the fragment. There, the link itself defines the type.

To make a collection of fragments actually become a fragment space, a set of rules is required determining which of the links are compatible with each other. Fragments that have compatible links can be connected via a new bond between the attachment points to form a larger fragment. Figure 9 shows an example of such a fragment collection. The lines connect fragments with compatible links. In case dummy link atoms are used, as is the case in this example, these will be removed upon fragment connection. The resulting fragment can contain additional links. This depends on the number of links in each of the two fragments that are to be connected. Furthermore, some additional specifications for the modification of the geometry upon fragment connection and the definition of terminal groups for link atoms may be necessary. The fragment space shown was generated by applying the RECAP method [176] to a subset of the WDI. This fragment space was used within the program TOPAS for the first time [261].

The probably most prominent example of a fragment space is a combinatorial library that fulfils all the essential criteria defined above (sees Section 4.2). Other examples for fragment spaces are the use of privileged molecular substructures [23, 24], cyclic and acyclic compounds [13], chemical functional groups [70, 251], building blocks [67], scaffolds and linkers [160], and single atoms [32, 227] or molecular fragments [204, 274].

5.2 *De Novo* Design Algorithms

As we have already outlined earlier, algorithms for docking and structure-based *de novo* design are closely related. Following the docking section above, we will summarize the methods used briefly.

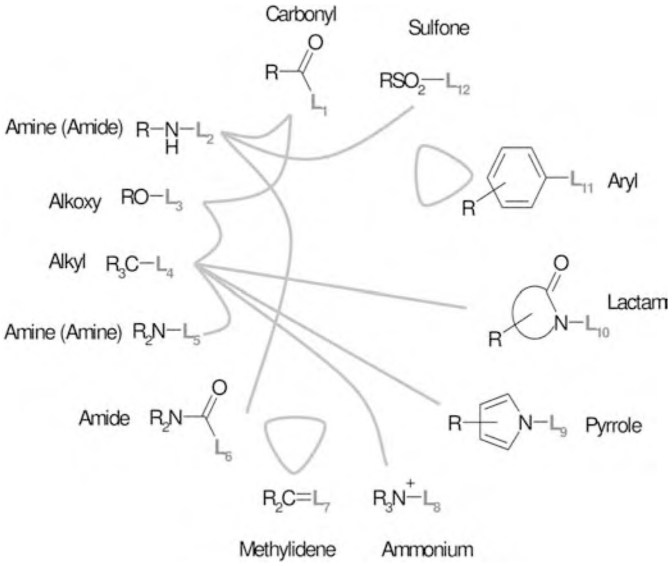


Figure 9 Fragment spaces. Collection of fragment prototypes contained in a fragment space. Each fragment has a (dummy) link atom that represents a corresponding chemical environment (L_1–L_{12}). Fragments can be connected via the formation of a bond between the atoms adjacent to the link atoms. The link atoms themselves will be removed upon fragment connection. The lines connecting the fragments indicate the compatibility of the link atoms. By this, the fragments span up the fragment space.

5.2.1 **Rigid-body Algorithms**

Lewis and Dean published one of the first approaches for *de novo* design more than 15 years ago in the late 1980s. They considered purely geometrical properties and used a collection of 2-D spacer skeletons and rigid-body transformations in order to explore the geometrical constraints of a receptor pocket [179]. The approach was extended to the third dimension in a subsequent publication [178]. Their program BUILDER uses docked molecules for the creation of regular or irregular molecular lattices. Thereon, linear atomic chains are constructed in a partly interactive manner [177, 180]. The amount of chemical knowledge, which could be taken into account for the first approaches, was very limited. That means that for the creation of the atomic chains, for example, only sp^3 carbon atoms were used. Therefore, the chemical models in the BUILDER program were extended in its second version [248].

5.2.2 **Simulation Methods**

One possible first step to take for a structure-based *de novo* design method is exactly the same as for docking calculations, i.e. one has to determine favorable interaction sites in a protein cavity. As mentioned previously, using different kinds of simulation methods can be used for this task. In the case of *de novo* design, the calculation can be carried out, for example, by a grid-based algorithm like in GRID [106], via a MD simulation as in MCSS [205] or by using a fragment-based docking method as done in SEED [189]. This information can then be used in subsequent computational steps. The DLD method, for example, searches a database for appropriate skeletons that can link several MCSS-derived minima [204]. There are a number of other programs that use GRID- or MCSS-derived minima as starting points [76, 170] (for a comparison, see Ref. [31]).

An alternative method is the following. Template structures are combined in a stepwise manner to create molecular skeletons, which satisfy the steric constraints of a given receptor. In a subsequent step, atom-type assignment is then carried out on these molecular skeletons to generate molecules that have complementary chemical properties to the active site. The atom assignment problem has been extensively examined in this context by, for example, Barakat and coworkers [15–19]. Programs based on this kind of approach are SPROUT [96–98, 136], which is based on graph-searching algorithms, and Skelgen [281, 282], which uses a MC-type method.

A third algorithmic alternative is to randomly place fragments in the active site, and assign atom types and connections between these fragments. This can be done by performing a MD simulation as in CONCEPTS [227] or CONCERTS [228], or by using a MC algorithm as implemented in MCDNLG [91], SMoG [70], GrowMol [32] or the method developed by Pellegrini and coworkers which replaces fragments of an already docked compound [230].

The characteristics of MD simulations are the same for *de novo* design as for screening large sets of molecules. Therefore, MD is not an appropriate method for exploring fragment spaces. Nevertheless, it can be of use for optimizing a limited number of solutions that were generated from a *de novo* design method. For the MC-type approaches, the statistical nature of the approach becomes even more apparent in the context of *de novo* design. Whereas this leads "only" to different sets of orientations of the ligand in the case of a docking calculation, for *de novo* design, sets of possibly totally different compounds are generated in each run. This makes it in most cases necessary to perform a multitude of calculations in order to obtain a diverse and probably more comprehensive set of solutions.

5.2.3 "Place & Join" Algorithms

In contrast to the corresponding docking approaches, the number of fragments available in a fragment space is obviously much larger than for a typical ligand molecule. Additionally, there is no information available concerning the topology and the size of the final compound(s). This is of course due to the nature of the fragment space, but has to be pointed out here, since this is a fundamental difference compared to docking methods. There are a couple of *de novo* design programs available which follow this strategy; all of them operate in a very similar way.

LUDI fits molecular fragments onto previously derived interaction sites in the protein cavity. In the first version, these start fragments were then connected by using bridge fragments from a database. Alternatively, LUDI can also be used for adding new substituents to an already existing ligand [33, 34, 38, 39]. Quite similar in concept are the following programs: CLIX [170], which uses fragment energy minima derived from GRID calculations, SPLICE [125], PRO_LIGAND [53, 54, 86, 212, 299, 301] and HOOK, which take fragments derived from MCSS minima as starting positions [76]. Leach and Kilvington used a "tweak" algorithm to generate families of acyclic linkers for combining a set of molecular fragments [173].

One advantage of "place and join" algorithms is that specific key interactions can be accounted for right from the beginning of the calculation. This ensures that all resulting compounds have this specific interaction pattern. However, the main difficulty of all the programs described above is the same as for the corresponding docking approaches. The compounds resulting from the bridging of multiple fragments within an active site are very much dependent on the amount and type of bridge fragments available. In addition, the probability of obtaining strained conformations is relatively high, which means that in most cases an additional energy minimization of the final compounds is necessary. This is then of course more costly in terms of computation time and may also alter the initial fragment positions.

5.2.4 Sequential Growth Algorithms

The use of incremental construction algorithms has originally been invented for *de novo* design and was already covered in Section 3.2. Therefore, the main difference here is the same as already outlined for the "place & join" algorithms, i.e. that the amount of fragments available in a *de novo* design approach is much larger. In principle, two types of sequential growth methods for *de novo* design can be distinguished, those that use atoms as basic building blocks and those that use molecular fragments.

The atom-based programs often involve random elements concerning the selection of atom types and attachment points. In general, only a limited number of atom types, hybridization states and constraints for bond lengths and bond angles are taken into account. Programs using these kinds of approaches include LEGEND [214, 215], GenStar [250] and RASSE [185].

The fragment-based methods can be divided into programs that incrementally build up molecules and those that replace parts of an already docked molecule. Two representatives of the first category are GROW [204], which uses amino acid templates, and COREGEN [13], which uses a ring-linker-based assembly procedure. Programs belonging to the second category are, for example, GroupBuild [251], which replaces individual functional groups for a given compound, and PRO_SELECT [211], which uses a database approach to search for corresponding combinatorial chemistry replacements.

FlexNovo is a fragment-based molecular design program developed by our own group and belongs to the first category. It is based on the FlexX [242] molecular docking software, and therefore uses the same algorithms, chemical models and scoring functions. FlexNovo deals with large fragment spaces, incorporates the concept of pharmacophore type constraints and a large number of additional physicochemical property, geometry and diversity filters [68].

When using incremental construction algorithms, the probability of obtaining strained conformations in the end is probably lower compared to "place & join" algorithms. However, a difficulty that arises especially in the context of *de novo* design is that optimal intermediate solutions might not lead to optimal global solutions. Whereas this does not necessarily have to be the case for docking programs using these kinds of algorithms, which was already mentioned above, for *de novo* design programs the case is more complicated. There, not only the ranking of intermediate solutions for a single compound is important, rather a set of compounds each having totally different intermediate solutions has to be ranked accordingly. In the end, this often results in compounds that are not optimally positioned inside the active site or that lack some key interactions. This can be partially compensated by using pharmacophore type constraints to guide the construction more in the "right" direction, but is of course no guarantee for finding an optimal solution.

5.2.5 **Genetic Algorithms and Evolutionary Programming**

Nondeterministic approaches, such as genetic algorithms or evolutionary programming, have not been used in *de novo* design methods right from the beginning. They became popular approximately a decade ago and are now frequently used in quite a number of programs.

The main difference between all the *de novo* design methods based on these kinds of algorithms is the type of fitness function used for the ranking of solutions. Typically, this is a more or less standard energy function that accounts for the different energetic and geometric contributions to the total energy. Alternatively, scoring functions known from molecular docking programs can also be directly used. This is done for example by the ADAPT program [229], which uses the DOCK scoring function, or LEA3D [73], where the fitness evaluation is based on the FlexX scoring function.

In addition, the employed fragment spaces differ slightly, but not significantly. Glen developed a rather general genetic algorithm procedure that generates molecular structures under different types of constraints, for example a protein cavity, a pharmacophore or certain molecular properties [101]. In most cases, however, different types of molecular fragments are used as done, for example, in LigBuilder [297], ADAPT [229], PEP [45] and LEA3D [73].

Due to the nature of genetic algorithms, the outcome of different calculations will result in different solution sets. Therefore, often not a single, but a couple of program runs are necessary in order to obtain probably a more reasonable set of diverse results, which is especially important in the field of *de novo* design.

5.3 Synthetic Accessibility

Synthetic accessibility has always been and still is one of the key issues in the computational *de novo* design area as shown by Stahl and coworkers [274] and recently surveyed by Baber and Feher [14]. Obviously, this is not the only criterion that a proposed lead or drug candidate has to fulfill. There are a number of at least equally important prerequisites, such as pharmacological activity and ADME toxicology properties [175,236]. The coverage of the latter is clearly beyond the scope of this chapter and is described in Chapter 19.

One way to incorporate synthetic accessibility in a *de novo* design approach is to encode it directly in the fragment space. This can be done by carefully choosing the set of fragments and by using well-defined connection rules. The (justified) assumption here is that compounds created from this fragment space will have a high probability on being synthetically tractable [23,24,67, 171,233,261].

5.3.1 Fragment Selection

There are quite a number of publications that deal with the issue of fragment selection. Each of them tries to identify and select common molecular substructures or functional groups in known drugs. This has been done, for example, for molecular frameworks [23] and side-chains [24], privileged molecular fragments [176], frequently replaced chemical groups [264], multi-activity substructures in molecules which show different biological activities [263], and heterocycles [43].

A complementary approach is to select fragments according to specific property ranges. The underlying assumption here is that there is a drug-like subspace in the vastness of the chemical universe that can be described by a limited number of easy accessible physicochemical properties [117,182,218]. Obviously, not all of the approaches for property prediction and selection can be directly applied to fragments as well. Nevertheless, there are some publications, which deal especially with the problem of fragment properties. A couple of them will be described below.

Following Lipinski's "Rule of Five" [183] (see also Chapter 18), Congreve and coworkers proposed a "Rule of Three" for fragment-based lead discovery and indicated that these property ranges might be useful for fragment selection [60]. Vieth and coworkers performed a survey on characteristic physical properties and the corresponding structural fragments for marketed oral drugs [287]. Hopkins and coworkers suggested the term "ligand efficiency" as an estimate of the binding energy of a compound on a per atom basis [132], which makes this approach in principle also applicable for fragments. This concept has been recently reviewed and extended by Abad-Zapatero and Metz [4].

Additionally, the exclusion of specific substructures could be a desirable task. Baurin and coworkers formulated a medicinal chemistry tractability filter for reactive or unwanted chemical features [20, 21]. Kazius and coworkers derived a set of toxicophores for mutagenicity prediction [143]. These filters have to be handled with care though [162], but can be of use for fragment-based drug discovery in principle [80].

5.3.2 Virtual Synthesis

The definition of the compatibility of fragments is essential when working with fragment spaces. There are several approaches for performing virtual synthesis of molecules. Each of the *de novo* design methods described above has its own definition of fragment compatibility. This ranges from the construction of molecules on an atom-per-atom basis to the connection of very specific substructures. Therefore, the methods differ mainly with respect to the level of sophistication they use for the evaluation carried out upon the formation of new bonds. In the following, a couple of approaches will be described which perform virtual synthesis in a more elaborate and, therefore, in a probably more accurate way.

The compatibility of fragments generated by the RECAP method, for example, is defined following the same rules for the connection of fragments that were used for cleaving compounds in the first place [176]. A more sophisticated evaluation is performed by the program package WODCA [90] and especially in a specific part of this package, the EROS program. This approach incorporates more elaborate physicochemical concepts and therefore tries to model a broad scope of organic reactions [129]. SYNOPSIS simulates organic reaction steps in a different way. For each compound, appropriate functional groups are detected and then different reactions are chosen from a collection of 70 different reaction types [288].

5.3.3 Compound Analysis

The use of drug-like fragments and the definition of a drug-like property range can be beneficial for the outcome of a *de novo* design approach for obvious reasons, but it is by no means sufficient to guarantee the synthetic accessibility of a certain compound [162]. There are a couple of other methods available which take more information into account than just molecular substructures. For example, neural nets have been used for distinguishing between drug-like and nondrug-like compounds [6, 47, 253], Muegge and coworkers used a pharmacophore point filter [207], and Byvatov and coworkers and Müller and coworkers used a support vector machine approach [47, 210]. It has been shown that each of these methods can be used for classifying compounds according to their similarity to known drugs or another training set of compounds with reasonable accuracy (see also Chapter 18).

Another possibility is to do an exhaustive retrosynthetic analysis for a given compound. The first attempts in doing this with computers were published in 1969 by Corey and Wipke [63]. Since then, a couple of related approaches have been published. Again, the coverage of all the information is beyond the scope of this article, but this subject has been excellently reviewed recently by Hanessian [116]. Such programs can identify possible synthesis pathways and appropriate starting chemicals from compound catalogues. Due to the fact that these are knowledge-based approaches, the outcome is very much dependent on the chemistry actually implemented in the program and the level of sophistication with which the analysis is done. The outcome of such an analysis can be of great help in the decision making process. However, still, it seems that the issue of synthetic accessibility cannot be described in a simple and consistent manner, as was recently investigated by Lajiness [167].

6 Structure-based Drug Design at Work: Validation Studies and Applications

Every computational tool for structure-based drug design is published with an initial validation of the method. Basically, we have to distinguish retrospective from prospective validation. In a retrospective study, previously collected experimental data, either X-ray crystal structures or binding affinities, are used to design a test scenario. In a prospective study, the calculated structures or binding affinities are validated afterwards by experiment.

A necessary condition for the applicability of a docking algorithm is its ability to reproduce X-ray structures of the molecular complexes. Usually, a few tenths of these structures are considered and most docking tools are able to reproduce roughly 70% of them within an error margin of 2 Å RMSD. The selection of these test cases is not a simple task: structures may be inappropriate (covalently bound ligands, ligand binding influenced by crystal contacts, etc.), of low resolution or even contain errors. A very good collection of protein–ligand complexes can be found in Ref. [216] and an alternative set has been published in Ref. [158]. When considering these redocking experiments, one has to keep in mind that the protein structure is usually taken from the bound conformation which makes redocking of X-ray structures to an easier problem than performing true predictions.

Several comparison studies have been published with the last 5 years (see, e.g. Refs. [29, 46, 87, 146, 157]). These comparisons have to be read with care because the results are influenced by many hidden parameters like the composition of the test cases, the preparation of the ligand and especially the protein, the selection of the active site, and the personal experience with the software tools just to name a view (see Ref. [59] for an excellent review on fallbacks in comparison studies).

Since most docking tools are applied for virtual screening, the question of whether the programs are able to extract bioactive compounds from large libraries has to be addressed. Obviously, predicting the complex structures correctly is necessary; however, it is not sufficient for virtual screening. In particular, the scoring function plays a more dominant role in virtual screening. While in redocking the scoring function has to select the right pose only, in virtual screening it has to distinguish between different ligand molecules. In retrospective screening studies, a set of known actives is mixed with other molecules having similar physicochemical properties than the known active ones. Literature containing retrospective virtual screening analysis is given in Ref. [59]. Chapter 18 gives a comprehensive review of structure-based virtual screening applications.

The ultimate test for a molecular docking or *de novo* design engine is a prospective study. Here, the experiments for structure determination or binding affinity measurement are done after the prediction. There are excellent reviews summarizing practical applications and success stories of structure-based virtual screening [164]. Chapter 18 further discusses these applications. Concerning *de novo* design, the field is not examined and reviewed in such great detail, as is the case for docking or virtual screening. Nevertheless, there have been a couple of more comprehensive studies for the use of such methods in drug discovery [131, 160, 274].

7 Concluding Remarks

The development of computational approaches for the prediction of protein–ligand interactions has a history lasting more than 30 years. Retrospectively, we can associate key developments with every decade. The 1970s were dominated by the simulation approach. In the 1980s, the first algorithmic approaches were presented. At this time, it was firstly possible to perform screening, i.e. to prioritize a large collection of compounds instead of looking at individual compounds only. Most approaches presented at that time were rigid-body docking algorithms. In the 1990s, docking tools began to consider ligand flexibility while maintaining the efficiency necessary for virtual screening. In parallel, several methods and software tools for structure-based *de novo* design were developed. The question arises as to what the current decade, once behind us, will stand for?

With no doubt, the most challenging problems in molecular docking are the consideration of protein flexibility during ligand binding (modeling the induced fit) and an accurate prediction of free binding energies. First algorithmic approaches dealing with protein flexibility can be found in the literature, some of them are covered in this chapter. These algorithms are able to deal with special cases like side-chain flexibility. However, none of them has yet been applied successfully in a virtual screening exercise. So far, the increasing number of false positives is the major stumbling block. Concerning scoring functions, we notice a continuous development of new and improvement of existing scoring functions. However, none of these developments can be considered as a big step forward. So for both fields, no satisfactory solution is in sight and it remains questionable whether these problems will be solved soon.

Although the scientific achievements for these two major problems in structure-based drug design are small, we can surely say that currently a big step is being made concerning the technology, applicability and acceptance of structure-based molecular design techniques. While the end of the 1990s was dominated by interest in (experimental) high-throughput screening, nowadays we recognize great interest in virtual screening techniques. Virtual screening facilities were established in nearly every large pharmaceutical company. According to PubMed, in 2004 about 400 papers related to virtual screening were published, compared to 134 in 1999. *De novo* design is not yet an established procedure in pharmaceutical industry, but is considered a complementary technique. In particular, in cases where virtual screening fails, *de novo* design can be of use as a generator of ideas for the identification of new chemotypes or for increasing the potency of a known inhibitor. Despite known deficiencies of the existing methods, they prove to be useful and belong today to the methodical standard repertoire for drug design.

References

1 FRED. OpenEye Scientific Software, Santa Fe, NM. http://www.eyesopen.com/products/applications/fred.html

2 OMEGA. OpenEye Scientific Software, Santa Fe, NM. http://www.eyesopen.com/products/applications/omega.html

3 ROTATE. Molecular Networks, Erlangen. http://www.mol-net.de/software/rotate

4 ABAD-ZAPATERO, C. AND J. T. METZ. 2005. Ligand efficiency indices as guideposts for drug discovery. Drug Discov. Today **10**: 464–9.

5 Abagyan, R. and M. Totrov. 1994. Biased probability Monte Carlo conformational searches and electrostatic calculations for peptides and proteins. J. Mol. Med. **235**: 983–1002.

6 Ajay, W. P. Walters and M. A. Murcko. 1998. Can we learn to distinguish between “drug-like” and “nondrug-like” molecules? J. Med. Chem. **41**: 3314–24.

7 Allen, F. H., S. Bellard, M. D. Brice, et al. 1979. The Cambridge Crystallographic Data Centre: computer-based search, retrieval, analysis and display of information. Acta Crystallogr. **B35**: 2331–9.

8 Antes, I., C. Merkwirth and T. Lengauer. 2005. POEM: Parameter Optimization using Ensemble Methods: application to target specific scoring functions. J. Chem. Inf. Model. **45**: 1291–302.

9 Apaydin, M. S., D. L. Brutlag, C. Guestrin, D. Hsu and J.-C. Latombe. 2002. Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion, in Proceedings of the 6th Annual International Conference on Computational Biology: 12–21.

10 Apaydin, M. S., C. Guestrin, C. Varma, D. L. Brutlag and J.-C. Latombe. 2002. Stochastic roadmap simulation for the study of ligand–protein interactions. Bioinformatics **18**: 18–26.

11 Apostolakis, J., A. Plückthun and A. Caflisch. 1998. Docking small ligands in flexible binding sites. J. Comput. Chem. **19**: 21–37.

12 Åqvist, J., C. Medina and J. E. Samuelson. 1994. A new method for predicting binding affinity in computer-aided drug design. Protein Eng. **7**: 385–91.

13 Aranov, A. M. and G. W. Bemis. 2004. A minimalist approach to fragment-based ligand design using common rings and linkers: application to kinase inhibitors. Proteins **57**: 36–50.

14 Baber, J. C. and M. Feher. 2004. Predicting synthetic accessibility: application in drug discovery and development. Mini Rev. Med. Chem. **4**: 681–92.

15 Barakat, M. T. and P. M. Dean. 1995. The atom assignment problem in automated *de novo* drug design. 1. Transferability of molecular fragment properties. J. Comput. Aided. Mol. Des. **9**: 341–50.

16 Barakat, M. T. and P. M. Dean. 1995. The atom assignment problem in automated *de novo* drug design. 2. A method for molecular graph and fragment perception. J. Comput. Aided. Mol. Des. **9**: 351–8.

17 Barakat, M. T. and P. M. Dean. 1995. The atom assignment problem in automated *de novo* drug design. 3. Algorithms for optimization of fragment placement onto 3D molecular graphs. J. Comput. Aided. Mol. Des. **9**: 359–72.

18 Barakat, M. T. and P. M. Dean. 1995. The atom assignment problem in automated *de novo* drug design. 4. Tests for site-directed fragment placement based on molecular complementarity. J. Comput. Aided. Mol. Des. **9**: 448–56.

19 Barakat, M. T. and P. M. Dean. 1995. The atom assignment problem in automated *de novo* drug design. 5. Tests for envelope-directed fragment placement based on molecular similarity. J. Comput. Aided. Mol. Des. **9**: 457–62.

20 Baurin, N., F. Aboul-Ela, X. Barril, et al. 2004. Design and characterization of libraries of molecular fragments for use in NMR screening against protein targets. J. Chem. Inf. Comput. Sci. **44**: 2157–66.

21 Baurin, N., R. Baker, C. Richardson, et al. 2004. Drug-like annotation and dublicate analysis of a 23-supplier chemical database totalling 2.7 million compounds. J. Chem. Inf. Comput. Sci. **44**: 643–51.

22 Baxter, C. A., C. W. Murray, D. E. Clark, D. R. Westhead and M. D. Eldridge. 1998. Flexible docking using tabu search and an empirical estimate of binding affinity. Proteins **33**: 367–82.

23 Bemis, G. W. and M. A. Murcko. 1996. The properties of known drugs, 1. Molecular frameworks. J. Med. Chem. **39**: 2887–93.

24 Bemis, G. W. and M. A. Murcko. 1999. Properties of known drugs. 2. Side chains. J. Med. Chem. **42**: 5095–9.

25 Berman, H. M., J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov and P. E. Bourne. 2000. The Protein Data Bank. Nucleic Acid Research **28**: 235–42.

26 Bernstein, F. C., T. F. Koetzle, G. J. B. Williams, et al. 1977. The Protein Data Bank: a computer based archival file for macromolecular structures. J. Mol. Med. **112**: 535–42.

27 Beveridge, D. L. and F. M. Di Capua. **1989**. Free energy via molecular simulation: applications to chemical and biomolecular systems. Annu. Rev. Biophys. Biophys. Chem. **18**: 431–92.

28 Billeter, M., T. F. Havel and I. D. Kuntz. 1987. A new approach to the problem of docking two molecules: the Ellipsoid algorithm. Biopolymers **26**: 777–93.

29 Bissantz, C., G. Folkers and D. Rogman. 2000. Protein based virtual screening of chemical databases. 1. Evaluation of different docking/scoring combinations. J. Med. Chem. **43**: 4759–67.

30 Bitetti-Putzer, R., D. Joseph-McCarthy, J. M. Hogle and M. Karplus. 2001. Functional group placement in protein binding sites: a comparison of GRID and MCSS. J. Comput. Aided Mol. Des. **15**: 935–60.

31 Bohacek, R., C. Mcmartin, P. Glunz and D. H. Rich. 1999. GrowMol, a *de novo* computer program, and its application to thermolysin and pepsin: results of the design and synthesis of a novel inhibitor. Math. Appl. **108**: 103–14.

32 Böhm, H.-J. 1992. The computer program LUDI: a new method for the *de novo* design of enzyme inhibitors. J. Comput. Aided Mol. Des. **6**: 61–78.

33 Böhm, H.-J. 1994. The development of a simple empirical scoring function to estimate the binding constant for a protein–ligand complex of known three-dimensional structure. J. Comput. Aided Mol. Des. **8**: 243–56.

34 Böhm, H.-J. 1992. LUDI: rule-based automatic design of new substituents for enzyme inhibitor leads. J. Comput. Aided Mol. Des. **6**: 593–606.

35 Böhm, H.-J. 1998. Prediction of binding constants of protein ligands: a fast method for the priorization of hits obtained from *de novo* design or 3D database search programs. J. Comput. Aided Mol. Des. **12**: 309–23.

36 Böhm, H.-J. 1995. Site directed structure generation by fragment joining. Perspect. Drug Discov. Des. **3**: 21–33.

37 Böhm, H.-J. 1996. Towards the automatic design of synthetically accessible protein ligands: peptides, amides andpeptidomimetics. J. Comput. Aided Mol. Des. **10**: 265–72.

38 Böhm, H.-J. and M. Stahl. 2002. The use of scoring functions in drug design discovery applications. Rev. Comput. Chem. **18**: 41–87.

39 Böhm, H. J., D. W. Banner and L. Weber. 1999. Combinatorial docking and Combinatorial chemistry: design of potent non-peptide thrombin inhibitors. J. Comput. Aided Mol. Des. **13**: 51–6.

40 Brooijmans, N. and I. D. Kuntz. 2003. Molecular recognition and docking algorithms. Annu. Rev. Biophys. Biomol. Struct. **32**: 335–73.

41 Broughton, H. B. and I. A. Watson. 2004. Selection of heterocycles for drug design. J. Mol. Graph. Model. **23**: 51–8.

42 Buckingham, A. D. 1997. *Theoretical Treatments of Hydrogen Bonding*. Wiley, New York, NY.

43 Budin, N., S. Ahmed, N. Majeux and A. Caflisch. 2001. An evolutionary approach for structure-based design of natural and non-natural peptidic ligands. Comb. Chem. High Throughput Screen. **4**: 661–73.

44 Bursulaya, B. D., M. Totrov, R. Abagyan and C. L. Brooks, III. 2003. Comparative study of several algorithms for flexible ligand docking. J. Comput. Aided Mol. Des. **17**: 755–63.

45 Byvatov, E., U. Fechner, J. Sadowski and G. Schneider. 2003. Comparison of support vector machine and artificial neural network systems for drug/nondrug classification. J. Chem. Inf. Comput. Sci. **43**: 1882–9.

46 CAFLISCH, A. 1996. Computational combinatorial ligand design: application to human α-thrombin. J. Comput. Aided Mol. Des. **10**: 372–96.

47 CARBÓ, R. L., L. LEYDA AND M. ARNAU. 1980. How similar is a molecule to another? An electron density measure of similarity between two molecular structures. Int. J. Quant. Chem. **17**: 1185–9.

48 CECCHINI, M., P. KOLB, N. MAJEUX AND A. CAFLISCH. 2004. Automated docking of highly flexible ligands by genetic algorithms: a critical assessment. J. Med. Chem. **25**: 412–22.

49 CHARIFSON, P. S., J. J. CORKERZ, M. A. MURCKO AND W. P. WALTERS. 1999. Consensus scoring: a method for obtaining improved hit rates from docking databases of three-dimensional structures into proteins. J. Med. Chem. **42**: 5100–9.

50 CHEN, X., Z. L. JI, D. G. ZHI AND Y. Z. CHEN. 2002. CLiBE: a database of computed ligand binding energy for ligand–receptor complexes. Comput. Chem. **26**: 661–6.

51 CLARK, D. E., D. FRENKEL, S. A. LEVY, J. LI, C. W. MURRAY, B. ROBSON, B. WASZKOWYCZ AND D. R. WESTHEAD. 1995. PRO_LIGAND: an approach to *de novo* molecular design. 1. Application to the design of organic molecules. J. Comput. Aided Mol. Des. **9**: 13–32.

52 CLARK, D. E. AND C. W. MURRAY. 1995. PRO_LIGAND: an approach to *de novo* molecular design. 5. Tools for the analysis of generated structures. J. Chem. Inf. Comput. Sci. **35**: 914–23.

53 CLARK, K. P. AND AJAY. 1995. Flexible ligand docking without parameter adjustment across four ligand–receptor complexes. J. Comput. Chem. **16**: 1210–26.

54 CLARK, R. D., A. STRIZHEV, J. M. LEONARD, C. F. BLAKE AND J. B. MATTHEW. 2002. Consensus Scoring for ligand/protein interactions. J. Mol. Graph. Model. **20**: 281–95.

55 CLAUSSEN, H., C. BUNING, M. RAREY AND T. LENGAUER. 2001. FlexE: efficient molecular docking into flexible protein structures. J. Mol. Med. **308**: 377–95.

56 CLAUSSEN, H., C. BUNING, M. RAREY AND T. LENGAUER. 2001. Molecular docking into the flexible active site of aldose reductase using FlexE. In HÖLTJE H.-D. and W. SIPPL (eds.), *Rational Approaches to Drug Design: Proceedings of the 13th European Symposium on Quantitative Structure–Activity Relationships*. Prous Science, Barcelona: 324–33.

57 COLE, J. C., C. W. MURRAY, J. W. NISSINK, R. D. TAYLOR AND R. TAYLOR. 2005. Comparing protein–ligand docking programs is difficult. Proteins **60**: 325–32.

58 CONGREVE, M., R. CARR, C. MURRAY AND H. JHOTI. 2003. A "Rule of Three" for fragment-based lead discovery? Drug Discov. Today **8**: 867–77.

59 CONNOLLY, M. L. 1983. Analytical molecular surface calculation. J. Appl. Crystallogr. **16**: 548–58.

60 CONNOLLY, M. L. 1985. Molecular surface triangulation. J. Appl. Crystallogr. **18**: 499–505.

61 COREY, E. J. AND W. T. WIPKE. 1969. Computer-assisted design of complex organic synthesis. Science **166**: 178–92.

62 CORNELL, W. D., P. CIEPLAK, C. I. BAYLY, et al. 1995. A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. J. Am. Chem. Soc. **117**: 5179–97.

63 COUTSIAS, E. A., C. SEOK AND K. A. DILL. 2004. Using quaternions to calculate RMSD. J. Comput. Chem. **25**: 1849–57.

64 CRIPPEN, G. M. AND T. F. HAVEL. 1988. *Distance Geometry and Molecular Conformation*. Research Studies Press, Taunton.

65 CROSS, K. P., G. MYATT, C. YANG, M. A. FLIGNER, J. S. VERDUCCI AND P. E. J. BLOWER. 2003. Finding discriminating structural features by reassembling common building blocks. J. Med. Chem. **46**: 4770–5.

66 DEGEN, J. AND M. RAREY. 2005. FLEXNOVO: structure-based searching in large fragment spaces. ChemMedChem **1**: 854–68.

67 DesJarlais, R. L., R. P. Sheridan, J. S. Dixon, I. D. Kuntz and R. Venkataraghavan. 1986. Docking flexible ligands to macromolecular receptors by molecular shape. J. Med. Chem. **29**: 2149–53.

68 DeWitte, R. S. and E. I. Shakhnovich. 1996. SMoG: *de novo* design method based on simple, fast, and accurate free energy estimates. 1. Methodology and supporting evidence. J. Am. Chem. Soc. **118**: 11733–44.

69 Di Nola, A., D. Roccatano and H. J. Berendsen. 1994. Molecular dynamics simulation of the docking of substrates to proteins. Proteins **19**: 174–82.

70 Diller, D. J. and K. M. J. Merz. 2001. High throughput docking for library design and library prioritization. Proteins **43**: 113–24.

71 Douguet, D., H. Munier-Lehmann, G. Labesse and S. Pochet. 2005. LEA3D: a computer-aided ligand design for structure-based drug design. J. Med. Chem. **48**: 2457–68.

72 Dress, A. W. M. and T. F. Havel. 1988. Shortest path problems and molecular conformation. Discr. Appl. Math. **19**: 129–44.

73 Easthope, P. L. and T. F. Havel. 1989. Computational experience with an algorithm for tetrangle inequality bound smoothing. Bull. Math. Biol. **51**: 173–94.

74 Eisen, M. B., D. C. Wilez, M. Karplus and R. E. Hubbard. 1994. HOOK: a program for finding novel molecular architectures that satisfy the chemical and steric requirements of a macromolecular binding site. Proteins **19**: 199–221.

75 Eldridge, M. D., C. W. Murraz, T. R. Auton, G. V. Paolini and R. P. Mee. 1997. Empirical scoring functions: I. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. J. Comput. Aided Mol. Des. **11**: 425–45.

76 Ewing, T. J. A. and I. D. Kuntz. 1997. Critical evaluation of search algorithms for automated molecular docking and database screening. J. Comput. Chem. **18**: 1175–89.

77 Ewing, T. J. A., S. Makino, A. G. Skillman and I. D. Kuntz. 2001. DOCK 4.0: search strategies for automated molecular docking of flexible molecule databases. J. Comput. Aided Mol. Des. **15**: 411–28.

78 Fattori, D. 2004. Molecular recognition: the fragment approach in lead generation. Drug Discov. Today **9**: 229–38.

79 Feher, M., E. Deretey and S. Roy. 2003. BHB: a simple knowledge-based scoring function to improve the efficiency of database screening. J. Chem. Inf. Comput. Sci. **43**: 1316–27.

80 Ferrara, P., H. Gohlke, D. J. Price, G. Klebe and C. L. Brooks, III. 2004. Assessing scoring functions for protein–ligand interactions. J. Med. Chem. **47**: 3032–47.

81 Ferro, D. R. and J. Hermans. 1977. A different best rigid-body molecular fit routine. Acta Crystallogr. **A33**: 345–7.

82 Fischer, D., S. L. Lin, H. L. Wolfson and R. Nussinov. 1995. A geometry-based suite of molecular docking processes. J. Mol. Med. **248**: 459–77.

83 Fischer, D., R. Norel, R. Nussinov and H. J. Wolfson. 1993. 3D Docking of protein molecules. In Proc. 4th Annu. Combinatorial Pattern Matching Symp., Heidelberg: 20–34.

84 Frenkel, D., D. E. Clark, J. Li, C. W. Murray, R. O. B, B. Waszkowycz and D. R. Westhead. 1995. PRO_LIGAND: an approach to *de novo* molecular design. 4. Application to the design of peptides. J. Comput. Aided. Mol. Des. **9**: 213–25.

85 Friesner, R. A., J. L. Banks, R. B. Murphy, et al. 2004. Glide: a new approach for rapid, accurate docking and scoring. 1. Method and assessment of docking accuracy. J. Med. Chem. **47**: 1739–49.

86 Gallop, M. A., R. W. Barrett, W. J. Dower, P. A. Fodor and E. M. Gordon. 1994. Applications of combinatorial technologies to drug discovery. 1. Background and peptide combinatorial libraries. J. Med. Chem. **37**: 1233–251.

87 Gasteiger, J., C. Rudolph and J. Sadowski. 1990. Automatic generation of 3D-atomic coordinates for organic molecules. Tetrahedron Comput. Methodol. **3**: 537–47.

88 Gasteiger, J. P., M., M. Sitzmann, R. Höllering, O. Sacher, T. Kostka and N. Karg. 2000. Computer-assisted synthesis and reaction planning in combinatorial chemistry. Perspect. Drug Discov. Des. **20**: 245–6.

89 Gehlhaar, D. K., K. E. Moerder, D. Zichi, C. J. Sherman, R. C. Ogden and S. T. Freer. 1995. *De novo* design of enzyme inhibitors by Monte Carlo ligand generation. J. Med. Chem. **38**: 466–72.

90 Gehlhaar, D. K., G. M. Verkhivker, P. A. Rejto, C. J. Sherman, D. B. Fogel, L. J. Fogel and S. T. Freer. 1995. Molecular recognition of the inhibitor AG–1343 by HIV-1 protease: conformationally flexible docking by evolutionary programming. Chem. Biol. **2**: 317–24.

91 Geladi, P. and B. R. Kowalski. 1986. Partial least-squares regression: a tutorial. Anal. Chim. Acta **185**: 1–17.

92 Ghose, A. K. and G. M. Crippen. 1985. Geometrically feasible binding modes of a flexible ligand molecule at the receptor site. J. Comput. Chem. **6**: 350–9.

93 Gillet, V. J., A. P. Johnson, P. Mata, S. Sike and P. Williams. 1993. SPROUT: a program for structure generation. J. Comput. Aided Mol. Des. **7**: 127–53.

94 Gillet, V. J., G. Myatt, Z. Zsoldos and A. P. Johnson. 1995. SPROUT, HIPPO and CAESA: tools for *de novo* structure generation and estimation of synthetic accessibility. Perspect. Drug Discov. Des. **3**: 34–50.

95 Gillet, V. J., W. Newell, P. Mata, G. Myatt, S. Sike, Z. Zsoldos and A. P. Johnson. 1994. SPROUT: recent developments in the *de novo* design of molecules. J. Chem. Inf. Comput. Sci. **34**: 207–17.

96 Giordanetto, F., S. Cotesta, C. Catana, J.-V. Trosset, A. Vulpetti, P. W. F. Stouten and R. T. Kroemer. 2004. Novel scoring functions comprising QXP, SASA, and protein side-chain entropy terms. J. Chem. Inf. Comput. Sci. **44**: 882–93.

97 Given, J. A. and M. K. Gilson. 1998. A hierarchical method for generating low-energy conformers of a protein–ligand complex. Proteins **33**: 475–95.

98 Glen, R. C. and A. W. Payne. 1995. A genetic algorithm for the automated generation of molecules within constraints. J. Comput. Aided. Mol. Des. **9**: 181–202.

99 Gohlke, H., M. Hendlich and G. Klebe. 2000. Knowledge-based scoring function to predict protein–ligand interactions. J. Mol. Med. **295**: 337–56.

100 Gohlke, H. and G. Klebe. 2002. Approaches to the description and prediction of the binding affinity of small-molecule ligands to macromolecular receptors. Angew. Chem. Int. Ed. Engl. **41**: 2644–76.

101 Goldberg, D. E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley Longman, Reading, MA.

102 Goodford, P. J. 1985. A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. J. Med. Chem. **28**: 849–57.

103 Goodsell, D. S., G. M. Morris and A. J. Olson. 1996. Automated docking of flexible ligands: applications of AutoDock. J. Mol. Recog. **9**: 1–5.

104 Goodsell, D. S. and A. J. Olson. 1990. Automated docking of substrates to proteins by simulated annealing. Proteins **8**: 195–202.

105 Gordon, E. M., R. W. Barrett, W. J. Dower, P. A. Fodor and M. A. Gallop. 1994. Applications of combinatorial technologies to drug discovery. 2. Combinatorial organic synthesis, library screening strategies, and future directions. J. Med. Chem. **37**: 1386–401.

106 Green, D. V. S. 2003. Virtual screening of virtual libraries. Prog. Med. Chem. **41**: 61–97.

107 Griewel, A. and M. Rarey. 2005. Assessing optimization strategies for incremental construction molecular docking tools. In Torda, A., S. Kurtz and M. Rarey (eds.), Proceedings of the German Conference on Bioinformatics

GCB 2005, Gesellschaft für Informatik: 119–30.

108 GROOTENHUIS, P. D. J., D. C. ROE, P. A. KOLLMAN AND I. D. KUNTZ. 1994. Finding potential DNA-binding compounds by using molecular shape. J. Comput. Aided Mol. Des. **8**: 731–50.

109 GSCHWEND, D. A. AND I. D. KUNTZ. 1996. Orientational sampling and rigid-body minimization in molecular docking re visited: on-the-fly optimization and degeneracy removal. J. Comput. Aided Mol. Des. **10**: 123–32.

110 HALGREN, T. A., R. B. MURPHY, R. A. FRIESNER, H. S. BEARD, L. L. FRYE, W. T. POLLARD AND J. L. BANKS. 2004. Glide: a new approach for rapid, accurate docking and scoring. 2. Enrichment factors in database screening. J. Med. Chem. **47**: 1750–9.

111 HALPERIN, I., B. MA, H. WOLFSON AND R. NUSSINOV. 2002. Principles of docking: an overview of search algorithms and a guide to scoring functions. Proteins **47**: 409–43.

112 HANESSIAN, S. 2005. Man, machine and visual imagery in strategic synthesis planning: computer-perceived precursors for drug candidates. Curr. Opin. Drug Discov. Dev. **8**: 798–819.

113 HANN, M. M., A. R. LEACH AND G. HARPER. 2001. Molecular complexity and its impact on the probability of finding leads for drug discovery. J. Chem. Inf. Comput. Sci. **41**: 856–64.

114 HART, T. N. AND R. J. READ. 1992. A multiple-start Monte Carlo docking method. Proteins **13**: 206–22.

115 HAVEL, T. F., I. D. KUNTZ AND G. M. CRIPPEN. 1983. The combinatorial distance geometry approach to the calculation of molecular conformation I. A new approach to an old problem. J. Theor. Biol. **104**: 359–81.

116 HAVEL, T. F., I. D. KUNTZ AND G. M. CRIPPEN. 1983. The theory and practice of distance geometry. Bull. Math. Biol. **45**: 665–720.

117 HEAD, R. D., M. L. SMYTHE, T. I. OPREA, C. L. WALLER, S. M. GREEN AND G. R. MARSHALL. 1996. VALIDATE: a new method for the receptor-based prediction of binding affinities of novel ligands. J. Am. Chem. Soc. **118**: 3959–69.

118 HENDLICH, M., A. BERGNER, J. GÜNTHER AND G. KLEBE. 2003. Relibase: design and development of a database for comprehensive analysis of protein–ligand interactions. J. Mol. Med. **326**: 607–20.

119 HINDLE, S. A., M. RAREY, C. BUNING AND T. LENGAUER. 2002. Flexible docking under pharmacophore constraints. J. Comput. Aided Mol. Des. **16**: 129–49.

120 HINDLE, S. A., M. STAHL AND M. RAREY. 2003. Flexible docking under pharmacophore type constraints: application to virtual screening. In FORD M., D. LIVINGSTONE, J. DEARDEN AND H. VAN DE WATERBEEMD (eds.), *Designing Drugs and Crop Protectants: Processes, Problems and Solutions*. Blackwell, Boston, MA: 135–9.

121 HO, C. M. W. AND G. R. MARSHALL. 1993. SPLICE: a program to assemble partial query solutions from three-dimensional database searches into novel ligands. J. Comput. Aided Mol. Des. **7**: 623–47.

122 HODGKIN, E. E. AND W. G. RICHARDS. 1987. Molecular similarity based on electrostatic potential and electric field. Int. J. Quant. Chem. Quant. Biol. Symp. **14**: 105.

123 HOFFMANN, D., B. KRAMER, T. WASHIO, T. STEINMETZER, M. RAREY AND T. LENGAUER. 1999. Two-stage method for protein–ligand docking. J. Med. Chem. **42**: 4422–33.

124 HOFFMANN, D., T. WASHIO, K. GESSLER AND J. JACOB. 1998. Tackling concrete problems in molecular biophysics using Monte Carlo and related methods: glycosylation, folding, solvation. In Proc. Workshop on Monte Carlo Approach to Biopolymers and Protein Folding, Singapore: 153–70.

125 HÖLLERING, R., J. GASTEIGER, L. STEINHAUER, K.-P. SCHULZ AND A. HERWIG. 2000. Simulation of organic reactions: from the degradation of chemicals to combinatorial synthesis. J. Chem. Inf. Comput. Sci. **40**: 482–94.

126 Honig, A. and B. Nicholls. 1995. Classical electrostatics in biology and chemistry. Science **268**: 1144–9.

127 Honma, T. 2003. Recent advances in *de novo* design strategy for practical lead identification. Med. Res Rev. **23**: 606–32.

128 Hopkins, A. L., C. R. Groom and A. Alex. 2004. Ligand efficiency: a useful metric for lead generation. Drug Discov. Today **9**: 430–1.

129 Ishchenko, A. V. and E. I. Shakhnovich. 2002. Small Molecule Growth 2001 (SMoG2001): an improved knowledge-based scoring function for protein–ligand interactions. J. Med. Chem. **45**: 2770–80.

130 Jain, A. N. 1996. Scoring noncovalent protein–ligand interactions: a continuous differentiable function tuned to compute binding affinities. J. Comput. Aided Mol. Des. **10**: 427–40.

131 Jain, A. N. 2003. Surflex: fully automatic flexible molecular docking using a molecular similarity-based search engine. J. Med. Chem. **46**: 499–511.

132 Johnson, A. P., K. Boda, T. Lengyel and S. Weaver. 2004. Improved methods for the *de novo* design of synthetically accessible ligands. Presented at the 228th ACS National Meeting, Philadelphia, PA.

133 Jones, G., P. Willett and R. C. Glen. 1995. Molecular recognition of receptor sites using a genetic algorithm with a description of desolvation. J. Mol. Med. **245**: 43–53.

134 Jones, G., P. Willett, R. C. Glen, A. R. Leach and R. Taylor. 1997. Development and validation of a genetic algorithm for flexible docking. J. Mol. Med. **267**: 727–48.

135 Jorgensen, W. L., D. S. Maxwell and J. Tirado-Rives. 1996. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. J. Am. Chem. Soc. **118**: 11225–36.

136 Joseph-McCarthy, D., B. E. Thomas, IV, M. Belmarsh, D. Moustakas and J. C. Alvarez. 2003. Pharmacophore-based molecular docking to account for ligand flexibility. Proteins **51**: 172–88.

137 Kabsch, W. 1978. A discussion of the solution for the best rotation to relate two sets of vectors. Acta Crystallogr. **A34**: 827–28.

138 Kabsch, W. 1976. A solution for the best rotation to relate two sets of vectors. Acta Crystallogr. A**32**: 922–3.

139 Kazius, J., R. McGuire and R. Bursi. 2005. Derivation and validation of toxicophores for mutagenicity prediction. J. Med. Chem. **48**: 312–3 20.

140 Kearsley, S. K. and G. W. Smith. 1990. An alternative method for the alignment of molecular structures: maximizing electrostatic and steric overlap. Tetrahedron Computer Methodology **3**: 615–33.

141 Kearsley, S. K., D. J. Underwood, R. P. Sheridan and M. D. Miller. 1994. Flexibases: a way to enhance the use of molecular docking methods. J. Comput. Aided Mol. Des. **8**: 565–82.

142 Kellenberger, E., J. Rodrigo, P. Muller and D. Rognan. 2004. Comparative evaluation of eight docking tools for docking and virtual screening accuracy. Proteins **57**: 225–42.

143 Kellogg, G. E., J. C. Burnett and D. J. Abraham. 2001. Very empirical treatment of solvation and entropy: a force field derived from Log$P_{O/W}$. J. Comput. Aided Mol. Des. **15**: 381–93.

144 Kick, E. K., D. C. Roe, A. G. Skillman, L. Guangcheng, T. J. A. Ewing, Y. Sun, I. D. Kuntz and J. A. Ellman. 1997. Structure-based design and combinatorial chemistry yield low nanomolar inhibitors of cathepsin D. Chem. Biol. **4**: 297–307.

145 Kirkpatrik, S., C. D. J. Gelatt and M. P. Vecchi. 1983. Optimization by simulated annealing. Science **220**: 671–80.

146 Kitchen, D. B., H. Decornez, J. R. Furr and J. Bajorath. 2004. Docking and scoring in virtual screening for drug discovery: methods and applications. Nat. Rev. Drug Discov. **3**: 935–49.

147 Klebe, G. and T. Mietzner. 1994. A fast and efficient method to generate biologically relevant conformations. J. Comput. Aided Mol. Des. **8**: 583–606.

148 Klebe, G., T. Mietzner and F. Weber. 1994. Different approaches toward an automatic structural alignment of drug molecules: applications to sterol mimics, thrombin and thermolysin inhibitors. J. Comput. Aided Mol. Des. **8**: 751–78.

149 KNEGTEL, R. M. A., I. D. KUNTZ AND C. M. OSHIRO. 1997. Molecular docking to ensembles of protein structures. J. Mol. Med. **266**: 424–40.

150 KOLLMAN, P. A. 1993. Free energy calculations: applications to chemical and biochemical phenomena. Chem. Rev. **93**: 2395–417.

151 KONTOYIANNI, M., L. M. MCCLELLAN AND G. S. SOKOL. 2004. Evaluation of docking performance: comparative data on docking algorithms. J. Med. Chem. **47**: 558–65.

152 KRAMER, B., M. RAREY AND T. LENGAUER. 1999. Evaluation of the FlexX incremental construction algorithm for protein–ligand docking. Proteins **37**: 228–41.

153 KRAMMER, A., P. D. KIRCHHOFF, X. JIANG, C. M. VENKATACHALAM AND M. WALDMANN. 2005. LigScore: a novel scoring function for predicting binding affinities. J. Mol. Graph. Model. **23**: 395–407.

154 KRIER, M., J. X. DE ARAÚJO-JÚNIOR, M. SCHMITT, J. DURANTON, H. JUSTIANO-BASARAN, C. LUGNIER, J.-J. BOURGUIGNON AND D. ROGNAN. 2005. Design of small-sized libraries by combinatorial assembly of linkers and functional groups to a given scaffold: application to the structure-based optimization of a phosphodiesterase 4 inhibitor. J. Med. Chem. **48**: 3816–22.

155 KROVAT, E. M., T. STEINDL AND T. LANGER. 2005. Recent advances in docking and scoring. Curr. Comput. Aided Drug Design **1**: 93–102.

156 KUBINYI, H. 2003. Drug research: myths, hype and reality. Nat. Rev. Drug Discov. **2**: 665–68.

157 KUBINYI, H. 1998. Structure-based design of enzyme inhibitors and receptor ligands. Curr. Opin. Drug Discov. Dev. **1**: 4–15.

158 KUBINYI, H. 2006. Success stories of computer-aided drug design. In EKINS, S. (ed.), *Computer Applications in Pharmaceutical Research and Development*. Wiley, New York: 377–424.

159 KUHL, F. S., G. M. CRIPPEN AND D. K. FRIESEN. 1984. A combinatorial algorithm for calculating ligand binding. J. Comput. Chem. **5**: 24–34.

160 KUNTZ, I. D., J. M. BLANEY, S. J. OATLEY, R. L. LANGRIDGE AND T. E. FERRIN. 1982. A geometric approach to macromolecule–ligand interactions. J. Mol. Med. **161**: 269–88.

161 LAJINESS, M. S., G. M. MAGGIORA AND V. SHANMUGASUNDARAM. 2004. Assessment of the consistency of medicinal chemists in reviewing sets of compounds. J. Med. Chem. **47**: 4891–6.

162 LAMBERT, M. H. Docking conformationally flexible molecules into protein binding sites. In CHARIFSON, P. S. (ed.), *Practical Application of Computer Aided Drug Design*. Dekker, New York: 243–303.

163 LAMDAN, Y. AND H. WOLFSON. 1988. Geometric hashing: a general and efficient model-based recognition scheme. Proc. Int. Conf. Computer Vision: 237–49.

164 LAWRENCE, M. C. AND P. C. DAVIS. 1992. CLIX: a search algorithm for finding novel ligands capable of binding proteins of known three-dimensional structure. Proteins **12**: 31–41.

165 LAZAR, C., A. KLUCZYK, T. KIYOTA AND Y. KONISHI. 2004. Drug evolution concept in drug design: 1. Hybridisation method. J. Med. Chem. **47**: 6973–82.

166 LEACH, A. R. 1994. Ligand docking to proteins with discrete side-chain flexibility. J. Mol. Med. **235**: 345–56.

167 LEACH, A. R. AND S. R. KILVINGTON. 1994. Automated molecular design: a new fragment-joining algorithm. J. Comput. Aided Mol. Des. **8**: 283–98.

168 LEACH, A. R. AND I. D. KUNTZ. 1992. Conformational analysis of flexible ligands in macromolecular receptor sites. J. Comput. Chem. **13**: 730–48.

169 LEESON, P. D., A. M. DAVIS AND J. STEELE. 2004. Drug-like properties: guiding principles for design – or chemical prejudice? Drug Discov. Today Technologies **1**: 189–95.

170 Lewell, X. Q., D. B. Judd, S. P. Watson and M. M. Hann. 1998. RECAP-retrosynthetic combinatorial analysis procedure: a powerful technique for indentifying privileged molecular fragments with useful applications in combinatorial chemistry. J. Chem. Inf. Comput. Sci. **38**: 511–22.

171 Lewis, R. A. 1992. Automated site-directed drug design: a method for the generation of general three-dimensional molecular graphs. J. Mol. Graph. **10**: 131–43.

172 Lewis, R. A. 1990. Automated site-directed drug design: approaches to the formation of 3D molecular graphs. J. Comput. Aided Mol. Des. **4**: 205–10.

173 Lewis, R. A. and P. M. Dean. 1989. Automated site-directed drug design: the concept of spacer skeletons for primary structure generation. Proc. R. Soc. Lond. B **236**: 125–40.

174 Lewis, R. A., D. C. Roe, C. Huang, T. E. Ferrin, R. Langridge and I. D. Kuntz. 1992. Automated site-directed drug design using molecular lattices. J. Mol. Graph. **10**: 66–78.

175 Linnainmaa, S., D. Harwood and L. S. Davis. 1988. Pose determination of a three-dimensional object using triangle pairs. IEEE Trans. Pattern Anal. Mach. Intell. **10**: 634–46.

176 Lipinski, C. and A. Hopkins. 2004. Navigating chemical space for biology and medicine. Nature **432**: 855–61.

177 Lipinski, C. A., F. Lombardo, B. W. Dominy and P. J. Feeney. 1997. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Adv Drug Deliv. Rev. **23**: 3–25.

178 Lorber, D. M. and B. K. Shoichet. 1998. Flexible ligand docking using conformational ensembles. Protein Sci. **7**: 938–50.

179 Luo, Z., R. Wang and L. Lai. 1996. RASSE: a new method for structure-based drug design. J. Chem. Inf. Comput. Sci. **36**: 1187–94.

180 Luty, B. A., Z. R. Wasserman, P. F. W. Stouten, C. N. Hodge, M. Zacharias and J. A. McCammon. 1995. A molecular mechanics/grid method for evaluation of ligand–receptor interactions. J. Comput. Chem. **16**: 454–64.

181 Lyne, P. D. 2002. Structure-based virtual screening: an overview. Drug Discov. Today **7**: 1047–55.

182 MacKerell, J., D., D. Bashford, M. Bellot, et al. 1998. All-atom empirical potential for molecular modeling and dynamics studies of proteins. J. Phys. Chem. B **102**: 3586–616.

183 Majeux, N., M. Scarsi, J. Apostolakis, C. Ehrhardt and A. Caflisch. 1999. Exhaustive docking of molecular fragments with electrostatic solvation. Proteins **37**: 88–105.

184 Makino, S., T. J. A. Ewing and I. D. Kuntz. 1999. DREAM++: flexible docking program for virtual combinatorial libraries. J. Comput. Aided Mol. Des. **13**: 513–32.

185 Makino, S. and I. D. Kuntz. 1997. Automated flexible ligand docking method and its application for database search. J. Comput. Chem. **18**: 1812–25.

186 Mancera, R. L., P. Källablad and N. P. Todorov. 2004. Ligand–protein docking using a quantum stochastic tunneling optimization method. J. Comput. Chem. **25**: 858–64.

187 Mangoni, M., D. Roccatano and A. D. Nola. 1999. Docking of flexible ligands to flexible receptors in solution by molecular dynamics simulation. Proteins **35**: 153–62.

188 Marsden, P. M., D. Puvanendrampillai, J. B. O. Mitchell and R. C. Glen. 2004. Predicting protein–ligand binding affinities: a low scoring game? Org. Biomol. Chem. **2**: 3267–73.

189 McGann, M. R., H. R. Almond, A. Nicholls, J. A. Grant and F. K. Brown. 2003. Gaussian docking functions. Biopolymers **68**: 76–90.

190 McMartin, C. and R. S. Bohacek. 1997. QXP: Powerful, rapid computer algorithms for structure-based drug design. J. Comput. Aided Mol. Des. **11**: 333–44.

191 Mendez, R., R. Laplae, M. F. Lensink and S. Wodak. 2005. Assessment of

CAPRI Predictions in Rounds 3–5 shows progress in docking procedures. Proteins **60**: 150–69.

192 MENG, E. C., D. GSCHWEND, J. M. BLANEY AND I. D. KUNTZ. 1993. Orientational sampling and rigid-body minimization in molecular docking. Proteins **17**: 266–78.

193 MENG, E. C., I. D. KUNTZ, D. J. ABRAHAM AND G. E. KELLOGG. 1994. Evaluating docked complexes with the HINT exponential function and empirical atomic hydrophobicities. J. Comput. Aided Mol. Des. **8**: 299–306.

194 MENG, E. C., B. K. SHOICHET AND I. D. KUNTZ. 1992. Automated docking with grid-based energy evaluation. J. Comput. Chem. **13**: 505–24.

195 MERLITZ, H., B. BURGHARDT AND W. WENZEL. 2003. Application of the stochastic tunneling method to high throughput database screening. Chem. Phys. Lett. **370**: 68–73.

196 MERLITZ, H. AND W. WENZEL. 2002. Comparison of stochastic optimization methods for receptor ligand docking. Chem. Phys. Lett. **362**: 271–77.

197 MILLER, M. D., S. K. KEARSLEY, D. J. UNDERWOOD AND R. P. SHERIDAN. 1994. FLOG: a system to select "quasi-flexible" ligands complementary to a receptor of known three-dimensional structure. J. Comput. Aided Mol. Des. **8**: 153–74.

198 MIRANKER, A. AND M. KARPLUS. 1995. An automated method for dynamic ligand design. Proteins **23**: 472–90.

199 MIRANKER, A. AND M. KARPLUS. 1991. Functionality maps of binding sites: a multiple copy simultaneus search method. Proteins **11**: 29–34.

200 MITCHELL, J. B. O., R. A. LASKOWSKI, A. ALEX AND J. M. THORNTON. 1999. BLEEP – potential of mean force describing protein–ligand interactions: i. generating potential. J. Comput. Chem. **20**: 1165–76.

201 MITCHELL, T. M. 1997. *Machine Learning*. McGraw-Hill, Singapore.

202 MIZUTANI, M. Y., N. TOMIOKA AND A. ITAI. 1994. Rational automatic search method for stable docking models of protein and ligand. J. Mol. Med. **243**: 310–26.

203 MONTGOMERY, D. C. 1991. *Design and Analysis of Experiments*. Wiley, New York, NY.

204 MOON, J. B. AND W. J. HOWE. 1991. Computer design of bioactive molecules: a method for receptor-based *de novo* ligand design. Proteins **11**: 314–28.

205 MORRIS, G. M., D. S. GOODSELL, R. S. HALLIDAY, R. HUEY, W. E. HART, R. K. BELEW AND A. J. OLSON. 1998. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. J. Comput. Chem. **19**: 1639–62.

206 MORRIS, G. M., D. S. GOODSELL, R. HUEY AND A. J. OLSON. 1996. Distributed automated docking of flexible ligands to proteins: parallel applications of AutoDock 2.4. J. Comput. Aided Mol. Des. **10**: 293–304.

207 MUEGGE, I., S. L. HEALD AND D. BRITELLI. 2001. Simple selection criteria for drug-like chemical matter. J. Med. Chem. **44**: 1841–6.

208 MUEGGE, I. AND Y. C. MARTIN. 1999. A general and fast scoring function for protein–ligand interactions: a simplified potential approach. J. Med. Chem. **42**: 791–804.

209 MUEGGE, I. AND M. RAREY. 2001. Small molecule docking and scoring. Rev. Comput. Chem. **17**: 1–60.

210 MÜLLER, K.-R., G. RÄTSCH, S. SONNENBURG, S. MIKA, M. GRIMM AND N. HEINRICH. 2005. Classifying "drug-likeness" with kernel-based learning methods. J. Chem. Inf. Model. **45**: 249–53.

211 MURRAY, C., D. E. CLARK, T. R. AUTON, et al. 1997. PRO_SELECT: combining structure-based drug design and combinatorial chemistry for rapid lead discovery. 1. Technology. J. Comput. Aided Mol. Des. **11**: 193–207.

212 MURRAY, C. W., D. E. CLARK AND D. G. BYRNE. 1995. PRO_LIGAND: an approach to *de novo* molecular design. 6. Flexible fitting in the design of peptides. J. Comput. Aided Mol. Des. **9**: 381–95.

213 Muryshev, A. E., D. N. Tarasov, A. V. Butygin, O. Y. Butygina, A. B. Aleksandrov and S. M. Nikitin. 2003. A novel scoring function for molecular docking. J. Comput. Aided Mol. Des. **17**: 597–605.

214 Nishibata, Y. and A. Itai. 1991. Automatic creation of drug candidate structures based on receptor structure: starting point for artificial lead generation. Tetrahedron **47**: 8985–90.

215 Nishibata, Y. and A. Itai. 1993. Confirmation of usefulness of a structure construction program based on three-dimensional receptor structure for rational lead generation. J. Med. Chem. **36**: 2921–8.

216 Nissink, J. W. M., C. Murray, M. Hartshorn, M. L. Verdonk, J. C. Cole and R. Taylor. 2002. A new test set for validating predictions of protein–ligand interaction. Proteins **49**: 457–71.

217 Oostenbrink, C., A. Villa, A. E. Mark and W. F. van Gunsteren. 2004. A biomolecular force field based on the free enthalpy of hydration and solvation: the GROMOS force-field parameters sets 53A5 and 53A6. J. Comput. Chem. **25**: 1656–76.

218 Oprea, T. I. 2000. Property distribution of drug-related chemical databases. J. Comput. Aided Mol. Des. **14**: 251–64.

219 Oshiro, C. M., I. D. Kuntz and J. S. Dixon. 1995. Flexible ligand docking using a genetic algorithm. J. Comput. Aided Mol. Des. **9**: 113–30.

220 Ota, N. and D. A. Agard. 2001. Binding mode prediction for a flexible ligand in a flexible pocket using multi-conformation simulated annealing pseudo crystallographic refinement. J. Mol. Med. **314**: 307–17.

221 Ozrin, V. D., M. V. Subbotin and S. M. Nikitin. 2004. PLASS: Protein–Ligand Affinity Statistical Score – a knowledge-based force-field model of interaction derived from the PDB. J. Comput. Aided Mol. Des. **18**: 261–70.

222 Pang, Y.-P., E. Perola, K. Xu and F. G. Prendergast. 2001. EUDOC: a computer program for identification of drug interaction sites in macromolecules and drug leads from chemical databases. J. Comput. Chem. **22**: 1750–71.

223 Paul, N., E. Kellenberger, G. Bret, P. Muller and D. Rognan. 2004. Recovering the true targets of specific ligands by virtual screening of the protein data bank. Proteins **54**: 671–80.

224 Pearlman, D. A., D. A. Case, J. W. Caldwell, et al. 1995. AMBER, a package of computer programs for applying molecular dynamics, normal mode analysis, molecular dynamics and free energy calculations to simulate the structural and energetic properties of molecules. Comput. Phys. Commun. **91**: 1–41.

225 Pearlman, D. A. and P. S. Charifson. 2001. Are free energy calculations useful in practice? A comparison with rapid scoring functions for the p38 MAP kinase protein system. J. Med. Chem. **44**: 3417–23.

226 Pearlman, D. A. and P. S. Charifson. 2001. Improved scoring of ligand–protein interactions using OWFEG free energy grids. J. Med. Chem. **44**: 502–11.

227 Pearlman, D. A. and M. A. Murcko. 1993. CONCEPTS: new dynamic algorithm for *de novo* drug suggestion. J. Comput. Chem. **14**: 1184–93.

228 Pearlman, D. A. and M. A. Murcko. 1996. CONCERTS: dynamic connection of fragments as an approach to *de novo* ligand design. J. Med. Chem. **39**: 1651–63.

229 Pegg, S. C.-H., J. J. Haresco and I. D. Kuntz. 2001. A genetic algorithm for structure-based *de novo* design. J. Comput. Aided Mol. Des. **15**: 911–33.

230 Pellegrini, E. and M. J. Field. 2003. Development and testing of a *de novo* drug-design algorithm. J. Comput. Aided Mol. Des. **17**: 621–41.

231 Pérez, C. and A. R. Ortiz. 2001. Evaluation of docking functions for protein–ligand docking. J. Med. Chem. **44**: 3768–85.

232 Perola, E., W. P. Walters and P. S. Charifson. 2004. A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance. Proteins **56**: 235–49.

233 Pierce, A. C., G. Rao and G. W. Bemis. 2004. BREED: generating novel inhibitors

through hybridisation of known ligands. Application to CDK2, P38 and HIV protease. J. Med. Chem. **47**: 2768–75.

234 PLATZER, J. E. B., F. A. MOMANY AND H. A. SCHERAGA. 1972. Conformational energy calculations of enzyme–substrate interactions. Int. J. Peptide Protein Res. **4**: 201–19.

235 PRABHAKAR, P. AND A. M. KAYASTHA. 1994. Mechanism of DNA–drug interactions. Appl. Biochem. Biotechnol. **47**: 39–55.

236 PRITCHARD, J. F., M. JURIMA-ROMER, M. L. J. REIMER, E. MORTIMER, B. ROLFE AND M. N. CAYEN. 2003. Making better drugs: decision gates in non-clinical drug development. Nat. Rev. Drug Discov. **2**: 542–53.

237 RAREY, M., B. KRAMER, C. BERND AND T. LENGAUER. 1996. Time-efficient docking of similar flexible ligands. Pac. Symp. Biocomput. (electronic version at http://www.cgl.ucsf.edu/psb/psb96/-proceedings/eproceedings.html).

238 RAREY, M., B. KRAMER AND T. LENGAUER. 1999. Docking of hydrophobic ligands with interaction-based matching algorithms. Bioinformatics **15**: 243–50.

239 RAREY, M., B. KRAMER AND T. LENGAUER. 1997. Multiple automatic base selection: protein–ligand docking based on incremental construction without manual intervention. J. Comput. Aided Mol. Des. **11**: 369–84.

240 RAREY, M., B. KRAMER AND T. LENGAUER. 1999. The particle concept: placing discrete water molecules during protein–ligand docking predictions. Proteins **34**: 17–28.

241 RAREY, M., B. KRAMER AND T. LENGAUER. 1995. Time-efficient docking of flexible ligands into active sites of proteins. Proc. ISMB **3**: 300–08.

242 RAREY, M., B. KRAMER, T. LENGAUER AND G. KLEBE. 1996. A fast flexible docking method using an incremental construction algorithm. J. Mol. Biol. **261**: 470–89.

243 RAREY, M., C. LEMMEN AND H. MATTER. 2005. Algorithmic engines in virtual screening. In OPREA, T. I. (ed.), *Cheminformatics in Drug Discovery*, vol. 23. Wiley-VCH, Weinheim: 59–115.

244 RAREY, M. AND T. LENGAUER. 2000. A recursive algorithm for efficient combinatorial library docking. Perspect. Drug Discov. Des. **20**: 63–81.

245 RAREY, M., S. WEFING AND T. LENGAUER. 1996. Placement of medium-sized molecular fragments into active sites of proteins. J. Comput. Aided Mol. Des. **10**: 41–54.

246 RICHARDS, F. M. 1977. Areas, volumes, packing, and protein structure. Annu. Rev. Biophys. Bioeng. **6**: 151–76.

247 ROCHE, O., R. KIYAMA AND C. L. BROOKS, III. 2001. Ligand–Protein DataBase: linking protein–ligand complex structures to binding data. J. Med. Chem. **44**: 3592–598.

248 ROE, D. C. AND I. D. KUNTZ. 1995. BUILDER v.2: improving the chemistry of a *de novo* design strategy. J. Comput. Aided Mol. Des. **9**: 269–82.

249 ROGNAN, D., S. L. LAEUMOELLER, A. HOLM, S. BUUS AND V. TSCHINKE. 1999. Predicting binding affinities of protein ligands from three-dimensional models: application to peptide binding to class I major histocompatibility proteins. J. Med. Chem. **42**: 4650–8.

250 ROTSTEIN, S. H. AND M. A. MURCKO. 1993. GenStar: a method for *de novo* drug design. J. Comput. Aided Mol. Des. **7**: 23–43.

251 ROTSTEIN, S. H. AND M. A. MURCKO. 1993. GroupBuild: a fragment-based method for *de novo* drug design. J. Med. Chem. **36**: 1700–10.

252 SADOWSKI, J., J. GASTEIGER AND G. KLEBE. 1994. Comparison of automatic three-dimensional model builders using 639 X-ray structures. J. Chem. Inf. Comput. Sci. **34**: 1000–8.

253 SADOWSKI, J. AND H. KUBINYI. 1998. A scoring scheme for discriminating between drugs and nondrugs. J. Med. Chem. **41**: 3325–9.

254 SANDAK, B., R. NUSSINOV AND H. J. WOLFSON. 1994. 3-D flexible docking of molecules. In Proc. 1st IEEE Workshop on Shape and Pattern Recognition in Computational Biology. Seattle: 41–54.

255 SANDAK, B., R. NUSSINOV AND H. J. WOLFSON. 1995. An automated computer vision and robotics-based technique for 3D flexible biomolecular docking and matching. Comput. Appl. Biol. Sci. **11**: 87–99.

256 SANDAK, B., R. NUSSINOV AND H. J. WOLFSON. 1998. A method for biomolecular structural recognition and docking allowing conformational flexibility. J. Computat. Biol. **5**: 631–54.

257 SCHELLHAMMER, I. AND M. RAREY. 2004. FlexX-Scan: fast structure-based virtual screening. Proteins **57**: 504–17.

258 SCHERZLER, A. AND M. RAREY. 2005. Effizientes Virtuelles Screening durch Wiederverwertung redundanter Fragmentplatzierungen. Diploma thesis, University of Hamburg, Center for Bioinformatics.

259 SCHNECKE, V., C. A. SWANSON, E. D. GETZOFF, J. A. TAINER AND L. A. KUHN. 1998. Screening a peptidyl database for potential ligands to proteins with side-chain flexibility. Proteins **33**: 74–87.

260 SCHNEIDER, G. AND U. FECHNER. 2005. Computer-based *de novo* design of drug-like molecules. Nat. Rev. Drug Discov. **4**: 649–63.

261 SCHNEIDER, G., M.-L. LEE, M. STAHL AND P. SCHNEIDER. 2000. *De novo* design of molecular architectures by evolutionary assembly of drug-derived blocks. J. Comput. Aided Mol. Des. **14**: 487–94.

262 SCHULZ-GASCH, T. AND M. STAHL. 2004. Scoring functions for protein ligand interaction: a critical perspective. Drug Discov. Today Technologies **1**: 231–39.

263 SHERIDAN, R. P. 2003. Finding multiactivity substructures by mining databases of drug-like compounds. J. Chem. Inf. Comput. Sci. **43**: 1037–50.

264 SHERIDAN, R. P. 2002. The most common chemical replacements in drug-like compounds. J. Chem. Inf. Comput. Sci. **42**: 103–08.

265 SHOICHET, B. K. 2004. Virtual screening of chemical libraries. Nature **432**: 862–5.

266 SHOICHET, B. K., D. L. BODIAN AND I. D. KUNTZ. 1992. Molecular docking using shape descriptors. J. Comput. Chem. **13**: 380–97.

267 SHOICHET, B. K. AND I. D. KUNTZ. 1993. Matching chemistry and shape in molecular docking. Protein Eng. **6**: 723–32.

268 SIPPL, M. J. 1990. Calculation of conformational ensembles from potentials of mean force. An approach to the knowledge-based prediction of local structures in globular proteins. J. Mol. Med. **213**: 859–83.

269 SIPPL, M. J. AND H. STEGEBUCHNER. 1991. Superposition of three dimensional objects: a fast and numerically stable algorithm for the calculation of the matrix of optimal rotation. Comput Chem. **15**: 73–8.

270 SMELLIE, A. S., G. M. CRIPPEN AND W. G. RICHARDS. 1991. Fast drug-receptor mapping by site-directed distances: a novel method of predicting new pharmacological leads. J. Chem. Inf. Comp. Sci. **31**: 386–92.

271 SOBOLEV, V., R. C. WADE, G. VRIEND AND M. EDELMAN. 1996. Molecular docking using surface complementarity. Proteins **25**: 120–29.

272 STAHL, M. 2000. Modifications of the scoring function in FlexX for virtual screening applications. Perspect. Drug Discov. Des. **20**: 83–98.

273 STAHL, M. AND M. RAREY. 2001. Detailed analysis of scoring functions for virtual screening. J. Med. Chem. **44**: 1035–42.

274 STAHL, M., N. P. TODOROV, T. JAMES, H. MAUSER, H.-J. BÖHM AND P. M. DEAN. 2002. A validation study on the practical use of automated *de novo* design. J. Comput. Aided Mol. Des. **16**: 459–78.

275 STERNBERG, M. J. E., H. A. GABB AND R. M. JACKSON. 1999. Predictive docking of protein–protein and protein–DNA complexes. Curr. Opin. Struct. Biol. **8**: 250–56.

276 STOCKWELL, B. R. 2004. Exploring biology with small organic molecules. Nature **432**: 846–54.

277 SU, A. I., D. M. LORBER, G. S. WESTON, W. A. BAASE, B. W. MATTHEWS AND B. K. SHOICHET. 2001. Docking molecules by families to increase the diversity of hits in database screens: computational

strategy and experimental evaluation. Proteins **42**: 279–93.

278 Sun, Y., T. J. A. Ewing, A. G. Skillman and I. D. Kuntz. 1999. CombiDOCK: Structure-based combinatorial docking and library design. J. Comput. Aided Mol. Des. **12**: 597–604.

279 Takamatsu, Y. and A. Itai. 1998. A new method for predicting binding free energy between receptor and ligand. Proteins **33**: 62–73.

280 Thormann, M. and M. Pons. 2001. Massive docking of flexible ligands using environmental niches in parallelized genetic algorithms. J. Comput. Chem. **22**: 1971–82.

281 Todorov, N. P. and P. M. Dean. 1998. A branch-and-bound method for optimal atom-type assignment in *de novo* ligand design. J. Comput. Aided Mol. Des. **12**: 335–49.

282 Todorov, N. P. and P. M. Dean. 1997. Evaluation of a method for controlling molecular scaffold diversity in *de novo* ligand design. J. Comput. Aided Mol. Des. **11**: 175–92.

283 Trosset, J.-Y. and H. A. Scheraga. 1999. Flexible docking simulations: scaled collective variable Monte Carlo minimization approach using Bezier splines, and comparison with a standard Monte Carlo algorithm. J. Comput. Chem. **20**: 244–52.

284 Trosset, J.-Y. and H. A. Scheraga. 1999. PRODOCK: software package for protein modeling and docking. J. Comput. Chem. **20**: 412–27.

285 Verdonk, M. L., J. C. Cole, M. J. Hartshorn, C. W. Murray and R. Taylor. 2003. Improved protein–ligand docking using GOLD. Proteins **52**: 609–23.

286 Vieth, M., J. D. Hirst and C. L. Brooks III. 1998. Do active site conformations of small ligands correspond to low free-energy solution structures? J. Comput. Aided Mol. Des. **12**: 563–72.

287 Vieth, M., M. G. Siegel, R. E. Higgs, I. A. Watson, D. H. Robertson, K. A. Savin, G. L. Durst and P. A. Hipskind. 2004. Characteristic physical properties and structural fragments of marketed oral drugs. J. Med. Chem. **47**: 224–32.

288 Vinkers, H. M., M. R. de Jonge, F. F. D. Daeyaert, et al. 2003. SYNOPSIS: SYNthesize and OPtimize System *in Silico*. J. Med. Chem. **46**: 2765–73.

289 Wallqvist, A. and D. G. Covell. 1996. Docking enzyme–inhibitor complexes using a preference-based free-energy surface. Proteins **25**: 403–19.

290 Wang, J., P. A. Kollman and I. D. Kuntz. 1999. Flexible ligand docking: a multistep strategy approach. Proteins **36**: 1–19.

291 Wang, R., X. Fang, Y. Lu and S. Wang. 2004. The PDBind database: collection of binding affinities for protein–ligand complexes with known three-dimensional structures. J. Med. Chem. **47**: 2977–80.

292 Wang, R., X. Fang, Y. Lu, C. Y. Yang and S. Wang. 2005. The PDBbind database: methodologies and updates. J Med Chem **48**: 4111–9.

293 Wang, R., L. Lai and S. Wang. 2002. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. J. Comput. Aided Mol. Des. **16**: 11–26.

294 Wang, R., L. Liu and Y. Tang. 1998. SCORE: a new empirical method for estimating the binding affinity of a protein–ligand complex. J. Mol. Model. **4**: 379–94.

295 Wang, R., Y. Lu and S. Wang. 2003. Comparative evaluation of 11 scoring functions for molecular docking. J. Med. Chem. **46**: 2287–303.

296 Wang, R., Y. Xu, X. Fang and S. Wang. 2004. An extensive test of 14 scoring functions using the PDBbind refined set of 800 protein–ligand complexes. J. Chem. Inf. Comput. Sci. **44**: 2114–125.

297 Wang, R., G. Ying and L. Lai. 2000. LigBuilder: a multi-purpose program for structure-based drug design. J. Mol. Model. **6**: 498–516.

298 Wasserman, Z. R. and C. N. Hodge. 1996. Fitting an inhibitor into the active site of thermolysin: A molecular dynamics case study. Proteins **24**: 227–37.

299 Waszkowycz, B., D. E. Clark, D. Frenkel, J. Li, C. W. Murray, B. Robson and D. R. Westhead. 1994. PRO_LIGAND: an approach to *de novo* molecular design. 2. Design of novel molecules from molecular field analysis (MFA) models and pharmacophores. J. Med. Chem. **37**: 3994–4002.

300 Welch, W., J. Ruppert and A. N. Jain. 1996. Hammerhead: fast, fully automated docking of flexible ligands to protein binding sites. Chem. Biol. **3**: 449–62.

301 Westhead, D. R., D. E. Clark, D. Frenkel, J. Li, C. W. Murray, B. Robson and B. Waszkowycz. 1995. PRO_LIGAND: An approach to *de novo* molecular design. 3. A genetic algorithm for structure refinement. J. Comput. Aided Mol. Des. **9**: 139–48.

302 Williams, D. H. and B. Bardsley. 1999. Estimating binding constants – the hydrophobic effect and cooperativity. Perspect. Drug Discov. Des. **17**: 43–59.

303 Yang, J.-M. 2004. Development and evaluation of a generic evolutionary method for protein–ligand docking. J. Comput. Chem. **25**: 843–57.

304 Yue, S. 1990. Distance-constrained molecular docking by simulated annealing. Protein Eng. **4**: 177–84.

305 Zacharias, M. and H. Sklenar. 1999. Harmonic modes as variables to approximately account for receptor flexibility in ligand–receptor docking simulations: application to DNA minor groove ligand complex. J. Comput. Chem. **20**: 287–300.

306 Zhang, C., S. Liu, Q. Zhu and Y. Zhou. 2005. A knowledge-based energy function for protein–ligand, protein–protein, and protein–DNA complexes. J. Med. Chem. **48**: 2325–35.

307 Zhang, J., M. Aizawa, S. Amari, Y. Iwasawa, T. Nakano and K. Nakata. 2004. Development of KiBank, a database supporting structure-based drug design. Comput. Biol. Chem. **28**: 401–7.

17
Modeling Protein–Protein and Protein–DNA Docking

Andreas Hildebrandt, Oliver Kohlbacher and Hans-Peter Lenhof

1 Introduction

Since Paul Ehrlich formulated his famous principle *corpora non agunt nisi fixata* [meaning that a substance is not (biologically) active unless it is bound] at the beginning of the 20th century [32], physical interaction has been known as the key step in most biochemical processes. Direct physical interaction (*binding*) between an enzyme and a ligand is required as an initial step in metabolic reactions. Similarly, protein–protein interactions are crucial in signal transduction and protein–DNA interactions are the key events in gene regulation.

With the availability of a sufficient number of protein structures from X-ray crystallography in the 1980s it became feasible to predict the structure of protein–protein complexes from the structures of the unbound proteins. This prediction problem, known as the protein–protein docking problem (PPD), has since been tackled using a wide range of computational techniques. More recently, the intense research in protein–protein interaction networks has sparked renewed interest in docking techniques to predict or validate these networks from a structural perspective and to aid in the understanding of the fundamentals of these interactions. In addition, the understanding of protein–protein interactions and protein–DNA interactions is essential for the study of gene regulatory networks (GRNs). While most of the research in protein–protein docking is thus still basic research, biomedical applications are rapidly gaining interest. In contrast to protein–ligand docking, these applications are less direct. In particular with complex multicausal diseases the understanding of the interactions of potential targets with other proteins is pivotal during the target identification phase of modern drug design. Computational validation of regulatory and interaction networks can provide useful insights during this phase. Another application exists at the interface between protein–protein and protein–ligand docking. Small molecules inhibiting protein–protein association are hard to identify using ligand-docking approaches, because they usually do not bind to well-defined, deep binding pockets. Instead, association inhibitors can form their own binding sites in the protein interface

Bioinformatics - From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

[8]. Techniques from protein–protein docking might be more appropriate than techniques from protein–ligand docking to handle such cases.

Algorithms to model biomolecular interactions have to solve two problems: they have to predict both the *binding mode* (i.e. the relative orientation of the partners binding) and the *binding affinity* (i.e. their binding free energy ΔG) as precisely as possible. Obviously, these two goals go hand in hand: no accurate estimation of the binding free energy is possible if the binding mode is not correct and without a decent energy function the identification of the correct binding mode is impossible. Many algorithms address the first issue, the identification of the binding mode, based on the lock-and-key principle proposed by Emil Fischer in 1894 [39]. The lock-and-key principle implies a strong geometric complementarity of the binding partners. Rigid-body docking (RBD) algorithms exploit this principle to identify complementary regions of the two partners and thus deduce possible binding sites. Unfortunately, there are many examples where the molecules experience significant changes in their geometry upon binding – a mechanism known as *induced fit* [65]. This is one of the two most serious problems in protein docking today.

The second important issue in protein docking is the accurate estimation of the binding affinity. All docking algorithms employ a scoring function or energy function of some kind to identify good approximations of the "true" complex structure observed in nature *(true positives)* from incorrect complex structures (*false positives* or *decoys*). An ideal energy function would cleanly separate structures close to the true structure from these decoys. The problem of predicting the binding free energy turns out to be quite complex. It requires the modeling of numerous physical interactions, some of which are still not understood in their entirety. Particularly troublesome is the modeling of the effects of water on the biomolecular interactions. Water is responsible for the hydrophobic interaction and has a strong influence on the electrostatic interaction.

It is beyond the scope of this chapter to review every aspect of protein–protein and protein–DNA docking in detail. Instead, we want to focus on a spectrum of sophisticated docking techniques. Section 2 discusses protein–protein docking while Section 3 describes protein–DNA docking. For those readers who want to learn more, we want to point out several recent reviews. Two comprehensive reviews [103, 108] cover the majority of relevant techniques and a number of interesting new developments in protein–protein docking. Correlation-based protein–protein docking has been reviewed at length by Eisenstein and Katchalski-Katzir [33]. The review by Russell and coworkers [100] briefly discusses protein–protein docking in the context of protein–protein interactions.

2 Protein–Protein Interactions

2.1 Basic Concepts of Docking

Numerous methodologies have been developed to predict the three-dimensional (3-D) structure of a protein complex AB from the structures of its unbound components A and B. Most methods follow the general algorithmic scheme described below (see also Figure 1).

(i) *Preprocessing.* The 3-D structures of A and B are read from files. If the hydrogen atoms are not contained in the imported structure files, they are usually added to the structures. Depending on the method, atom radii and charges have to be assigned, and representations of the protein surfaces of A and B have to be calculated. The calculated molecular surfaces represent artificial boundaries between the interior and the exterior of the proteins.

(ii) *Generation of putative complex structures.* The majority of docking approaches are based on the assumption that the two proteins A and B undergo only limited conformational change during the docking process. RBD techniques treat the proteins A and B as completely rigid 3-D objects. Usually, an RBD approach scans the conformation space and generates a very large number of putative complex conformations (*candidates*) that are stored in a list. If the complex generation procedure is successful, the candidate list contains a sufficient number of true positives. Candidate structures are often generated by matching procedures that map regions of the surface of B onto geometrically and chemically 'compatible' regions of the surface of A.

(iii) *Filtering steps.* The generated putative complex conformations are evaluated with respect to shape and chemical complementarity using scoring functions. Candidate conformations with low scores are removed from the candidate list. The filtering procedure can be repeated with different scoring functions, starting with functions that can be computed efficiently. Apart from geometric and energetic scoring functions, distance constraints and other experimental data of the complex structure can be used to filter out decoys.

(iv) *Realizing protein flexibility.* RBD approaches usually succeed in the correct prediction of the protein complexes if the bound structures of A and B are sufficiently close to their unbound, native structures. However, small conformational changes involving only a few side-chains can cause RBD approaches to fail. In order to realize protein flexibility, docking algorithms thus optimize the conformations of the remaining candidates.

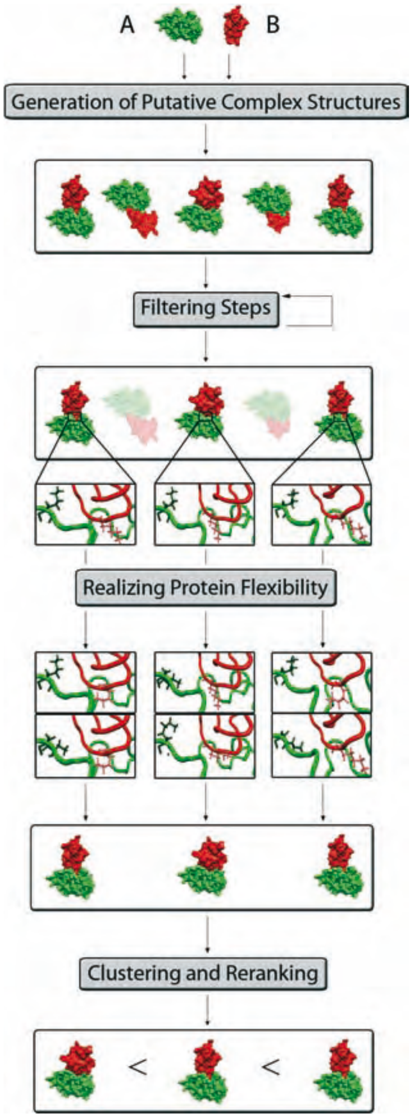

A
B
Generation of Putative Complex Structures
Filtering Steps
Realizing Protein Flexibility
Clustering and Reranking
<
<

Figure 1 Schematic overview of the major steps in protein–protein docking algorithms.

Some algorithms account for the flexibility of the amino acid side-chains at the binding site or of backbone hinges only, while other approaches optimize the coordinates of all atoms in the complex.

(v) *Clustering and re-ranking.* Candidate lists produced by docking algorithms often include families of similar structures. Consequently, many algorithms include a clustering step to reduce redundancy. In the last step, docking algorithms re-rank the remaining candidates or clusters of conformations using a more sophisticated energy or scoring function and return the best conformations.

The above scheme is only a broad outline of a typical docking algorithm. Individual algorithms can lack some of these components or differ in the underlying techniques. Key differences often are the structure generation, the scoring functions and the models of flexibility employed.

When comparing different algorithms, the two most obvious performance measures are "quality of the result" and "computational speed". However, the evaluation of the quality of docking results is almost as challenging as the docking problem itself. Typically, docking approaches are validated using X-ray or nuclear magnetic resonance structures of protein complexes. The algorithms usually output large lists of putative complex conformations sorted with respect to some scoring function. Successful algorithms will place true positives at the top of the list.

One widely used quality measure is the rank of the *first* true positive found in the candidate list (the lower the rank, the better the performance). Whether a structure is a true positive can be decided via structural similarity measures, e.g. the root-mean-square deviation (RMSD), or the fraction of native contacts formed in the protein–protein interface. A second important measure is the similarity of the *best true positive* contained in the list.

The performance of docking algorithms can be evaluated based on the bound structures of the proteins A and B or on their native structures. Both types of tests use protein complexes whose structures have been elucidated experimentally. In the first kind of tests, which we will call *bound tests* or *redocking*, the docking problem is simplified by ignoring all conformational changes that take place during the docking process. Here, the bound structures of A and B are taken from the protein complex AB, a random rigid transformation t is applied to one of the two proteins, e.g. to B, and the docking algorithm to be tested is then run on the structures A and $t(B)$. Bound tests are the simplest of all tests. Algorithms performing poorly in these tests will clearly fail in real-life examples as well. The RBD techniques presented in Section 2.2 that treat the proteins A and B as entirely rigid 3-D objects usually show excellent performance in bound tests.

In the second, more realistic kind of tests, which we will call *native test*, the algorithms are applied to native structures of A and B that may differ considerably from the bound structures. Since even small conformational changes can lead to the failure of RBD approaches, numerous techniques for realizing protein flexibility have been developed. A selection of sophisticated approaches for modeling protein flexibility will be presented in Section 2.3.

A series of "blind" prediction experiments with the goal to identify the strengths and weaknesses of existing docking approaches was initiated in 2001/2002. In these blind tests, docking algorithms have been applied to protein complexes whose structures have been elucidated, but not yet published. The results of the so-called CAPRI (Critical Assessment of PRediction of Interactions) competitions will be discussed in Section 2.6. These experiments have revealed that one of the key problems of the docking algorithms is the choice of an appropriate scoring function. A spectrum of widely used scoring functions will be discussed in Section 2.4. The integration of experimental data into docking algorithms is a way to improve the quality and reliability of docking results. In Section 2.5, we will present data-driven approaches to protein docking.

2.2 Rigid Body Docking

RBD approaches are based on Emil Fischer's lock-and-key principle, which implies local shape complementarity of the proteins at the binding site. In the case of RBD, it is assumed that the two proteins *A* and *B* show only a limited conformational change during the docking process. They can thus be treated as rigid 3-D objects.

Typically, RBD algorithms try to identify conformations with large contact areas between the surfaces of *A* and *B* that exhibit no or only small overlaps between the interiors. In order to test this condition for given candidates, we have to define and calculate artificial protein surfaces that allow for differentiation between the interior, the surface and the exterior of the proteins.

Almost any kind of established surface definition has been used for this purpose. In particular, van der Waals, solvent-excluded and solvent-accessible surfaces have been employed in different representations. Typically, RBD approaches try to identify putative complex conformations by keeping one protein fixed in space while moving the second around. The resulting candidates can then be uniquely represented by six parameters describing the relative translation and rotation of the mobile protein from the reference position. In the following, we will generally denote the static protein by *A* and the mobile protein by *B*.

2.2.1 Correlation Techniques

An important type of RBD docking techniques, so-called *correlation docking*, was introduced by Katchalski-Katzir and coworkers [58]. In their classical formulation, correlation-based docking algorithms use a purely geometric score to generate conformations featuring a large number of surface contacts and no significant overlap.

In order to identify candidates with this property, the molecular geometries are represented by piecewise constant functions on a 3-D grid (Figure 2). Let $\vec{r}_{ijk} \in \mathbb{R}^3$, $i, j, k \in \{1, \ldots, N\}$ denote a set of evenly spaced points in a box containing the molecules A and B, where $\vec{r}_{111}$ corresponds to the "lower left front" corner of the box and $\vec{r}_{NNN}$ to the "upper right back" corner.

The structures of A and B can be represented by two grids $\mathcal{A}$ and $\mathcal{B}$ with values a_{ijk} and b_{ijk} that depend on whether the point $\vec{r}_{ijk}$ is on the inside, the surface (to achieve a certain softness of the molecules, the surface is represented by a surface layer of finite thickness) or the outside of the respective molecule:

$$a_{ijk} = \begin{cases} 1 & \text{on the surface of molecule A} \\ \rho & \text{inside molecule A} \\ 0 & \text{outside molecule A} \end{cases} \tag{1}$$

$$b_{ijk} = \begin{cases} 1 & \text{on the surface of molecule B} \\ \delta & \text{inside molecule B} \\ 0 & \text{outside molecule B.} \end{cases} \tag{2}$$

At each point $\vec{r}_{ijk}$, the local fit of A and B in their current relative orientation can then be assessed by computing the score:

$$c_{ijk} := a_{ijk} b_{ijk}$$

and the geometric fitness value (Score) for a conformation represented by the grids $\mathcal{A}$ and $\mathcal{B}$ is computed as:

$$\text{Score} := \sum_{i,j,k=1}^{N} c_{ijk}$$

Depending on the shape, location and relative orientation of A and B, c_{ijk} can assume the values given in Table 1. Thus, each point on the surfaces of both molecules contributes with a value of 1 to the total score. To penalize overlap between A and B, ρ is set to a large negative value and δ to a small positive one, such that $\delta\rho$ results in a negative contribution to the geometric fitness for each point of overlap. Tuning the values of δ and ρ allows to adjust

Table 1

	Surface point of A	Interior point of A	Outside A
Surface point of B	1	$\rho \ll 0$	0
Interior point of B	$\delta > 0$	$\rho\,\delta \ll 0$	0
Outside B	0	0	0

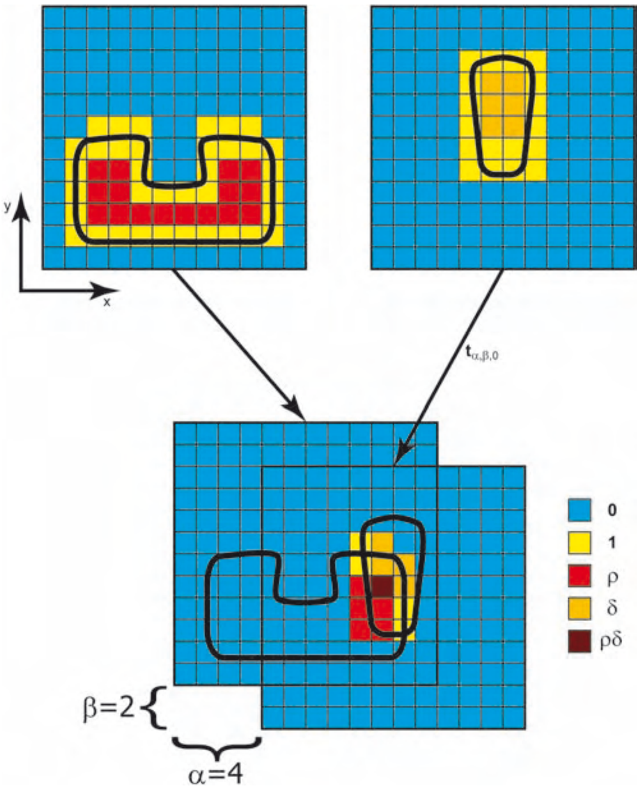


Figure 2 Schematic illustration of the grids used for correlation-based docking. First, each protein is mapped onto a grid describing its inside, outside, and surface. Afterwards, the two grids are correlated to determine the geometric fit between the two proteins in their current relative orientation.

for the degree of penetration that will be tolerated. This choice also strongly discourages the contact of the surface points of *B* and interior points of *A* with a contribution of $\rho \ll 0$, while penetration of *A*'s surface into the interior of *B* is allowed. This can be considered as an additional "softening" of the molecular surfaces.

To determine the complex structure leading to the largest geometric score, the larger of the two molecules, *A*, is kept fixed in its position while the smaller one, *B*, is rotated and translated to sample the configuration space. Correlation-based approaches only consider a discrete subset of the possible rotations of molecule *B*. For each of these possible rotations of *B*, the algorithms try to identify the optimal translations. Since the molecules themselves are represented by discrete grids, it is natural to restrict the translational degree of freedom to integral multiples of the grid spacing, i.e. given a certain rotation of molecule *B*, instead of scanning the whole translational space for the optimal conformation, we only test N^3 translations of molecule B, which are of the form $t_{\alpha\beta\gamma} := (\alpha\Delta_x, \beta\Delta_y, \gamma\Delta_z)$, where $\Delta_x, \Delta_y, \Delta_z$ denote the grid spacing of a_{ijk} and b_{ijk} in the x, y, and z directions. To simplify the notation, we will assume that $\Delta := \Delta_x = \Delta_y = \Delta_z$.

Translating the molecule B by a transformation $t_{\alpha\beta\gamma}$, with $\alpha, \beta, \gamma \in \mathbb{Z}$, corresponds to shifting the grid indices of $\mathcal{B}$. This implies that it is *not* necessary to introduce a different grid for each translation of molecule B. Instead, it suffices to read off the values from the non-translated grid $\mathcal{B}$ at the shifted position as follows:

$$t_{\alpha\beta\gamma}(b_{ijk}) = b_{i+\alpha,j+\beta,k+\gamma}$$

Thus, for each rotated conformation of molecule B, we can compute a grid of geometric scores for each of its translations $t_{\alpha\beta\gamma}$ from:

$$c_{\alpha\beta\gamma} = \sum_{i,j,k=1}^{N} a_{ijk} b_{i+\alpha,j+\beta,k+\gamma}$$

The values $c_{\alpha\beta\gamma}$ for all translations of B build a grid of scores which we denote by $\mathcal{C}$.

Let $\mathcal{R}$ be the set of all possible rotations of molecule B. The general scheme of correlation-based docking algorithms then looks as follows [33]:

- Compute the grid $\mathcal{A}$ for molecule A
- For each rotation $r \in \mathcal{R}$:
  - apply r to molecule B
  - compute the grid $\mathcal{B}$ corresponding to the current rotation state of B
  - compute the grid $\mathcal{C}$
  - store the m largest values of $\mathcal{C}$, corresponding to the m translations of the current rotation state of B with the largest geometric scores
- Sort the resulting $m|\mathcal{R}|$ conformations with respect to their scores and return them.

A direct naive implementation of this algorithm would result in $\mathcal{O}(N^3)$ time for each value $c_{\alpha\beta\gamma}$, leading to a total of $\mathcal{O}(N^6)$ time for the computation of $\mathcal{C}$ and a total of $\mathcal{O}(|\mathcal{R}|N^6)$ for the complete algorithm.

Fortunately, the runtime can be drastically decreased by noting that the grid $c_{\alpha\beta\gamma}$ is the discrete correlation of the grids $\mathcal{A}$ and $\mathcal{B}$. It can thus be computed efficiently using the fast Fourier transform (FFT) due to the *correlation theorem*:

$$\mathcal{FT}\left(\sum_{i,j,k=1}^{N} a_{ijk} b_{i+\alpha,j+\beta,k+\gamma}\right) = \mathcal{FT}^*(a_{ijk})\mathcal{FT}(b_{ijk})$$

Here, $\mathcal{FT}$ denotes the Fourier transform and $\mathcal{FT}^*$ its complex conjugate. This yields

$$c_{\alpha\beta\gamma} = \mathcal{FT}^{-1}(\mathcal{FT}^*(a_{ijk})\mathcal{FT}(b_{ijk}))$$

where $\mathcal{FT}^{-1}$ denotes the inverse Fourier transform.

Since the Fourier transform of a discrete grid of size N^3 can be computed in $\mathcal{O}(N^3 \log(N))$ time using the FFT, we can thus compute $\mathcal{C}$ in $\mathcal{O}(N^3 \log(N))$ time, leading to the total runtime of $\mathcal{O}(|\mathcal{R}|N^3 \log(N))$.

Even though this algorithm generates candidate structures based on *geometric* matching alone and thus neglects energetic aspects like electrostatic compatibility, it works quite well in *bound* docking experiments. A variety of improvements to this scheme has been proposed in the literature. In principle, two different techniques can be used to improve the scoring of the candidate structures. The first option is the re-evaluation of the candidate structures using more sophisticated scoring functions. In this case it is important to ensure that the initial geometry-based docking algorithm produces a sufficient number of near-native structures. A typical example of this approach is the docking programm BiGGER by Palma and coworkers [95], where electrostatic energy and desolvation are used as post-scan filters.

The second option is the modification of the filtering process to include energetic contributions in addition to geometric complementarity. A straightforward approach to this problem introduces additional grids. Gabb and coworkers [40] employ one grid for the electrostatic potential of protein A and one grid for the partial charges of B to obtain an estimate of the electrostatic energy of the complex, and use the result to filter out conformations with unfavorable interaction energies. Mandell and coworkers [81] propose a similar technique, but compute the potential of A using continuum electrostatics. The resulting interaction energy estimates are then added to the geometric fit of the candidate. An elegant alternative to introducing additional grids was pointed out by Heifetz and coworkers [48], who use grids of *complex* numbers and store the non-geometric information in the imaginary part. Rather than correlating electrostatic potentials and charges to obtain an estimate of the electrostatic interaction energies, they make use of the observed anticorrelation of the electrostatic potentials of docking partners at the binding site [82]. Similarly, hydrophobic interactions can be accounted for, e.g. in the imaginary part of $\mathcal{C}$. For a recent review on correlation-based docking techniques, see Ref. [33].

2.2.2 Graph-based Structure Generation Methods

In contrast to grid-based correlation techniques, graph-based approaches allow for accurate but compact representations of the protein structures or

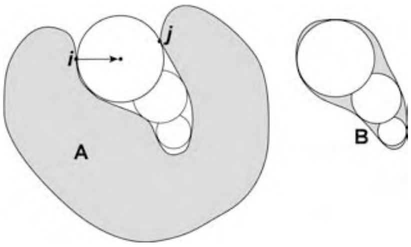


Figure 3 DOCK represents the binding pockets of protein A and the interior of protein B by sets of spheres and searches for good superpositions of these sets.

their surfaces. Kuntz and coworkers [66,105,106] developed DOCK (see also Chapter 16), a method that describes the geometry of the putative binding sites of A and the shape of B by sets of spheres. The union of spheres that belong to a pocket of A represents the empty volume of this putative binding site and builds a negative image of A at that site. The DOCK approach tries to identify the best superposition of B and the negative images of A by searching for cliques in a graph representing all possible matchings of spheres.

DOCK starts by computing discrete sets of points on the molecular surfaces of A and B. In the second step, the algorithm computes sets of spheres for A and B with the following properties (Figure 3):

(i) Each sphere touches the molecular surface at two points (i, j) and has its center on the surface normal of point i.

(ii) Each sphere of protein A lies on the outside of the surface of A.

(iii) Each sphere of protein B lies on the inside of the surface of B.

This procedure usually generates a large number of spheres, which can be reduced by applying additional constraints. The remaining spheres are clustered into “small” sets. Ideally, the sphere sets of A should represent the different pockets and tentative binding sites on the surface of A, while the clusters of B should form the set of possible docking interfaces of B.

The actual structure generation can be formulated as a *clique* (a completely connected subgraph) search problem in a graph that represents the putative pairwise matchings of spheres of A and B as nodes. For each pair (i, j), where i is a sphere of A and j is a sphere of B, a node v_{ij} is added to the graph. Two nodes v_{ij} and v_{kl} are connected by an edge if and only if the Euclidean distance $d(i, k)$ of the sphere centers i and k belonging to A is almost equal to the Euclidean distance $d(j, l)$ of the sphere centers j and l of B, i.e.:

$$\|d(i,k) - d(j,l)\| \leq \varepsilon \tag{3}$$

where ε is a user-defined parameter. Hence, edges connect pairs of spheres whose centers have almost the same distances and hence can be mapped onto each other by a suitable rigid transformation. Each clique of size four in the graph defines a possible transformation. Therefore, the algorithm calculates the tentative complex conformations by carrying out a straightforward incremental search for cliques of size four, by computing the respective transformations, and by applying the transformations to all atoms of protein B.

Kasinos and coworkers [57] presented another graph-based approach that applies graph-matching techniques to identify tentative complex conformations. They calculate points on the surfaces of A and B and two graphs that represent the surfaces. The points on the surfaces build the vertex sets of the respective graphs. Pairs of points are connected by edges that are labeled with the Euclidean distances between the points. The algorithm calculates maximal matchings of the two graphs and the transformations that map the respective point sets onto each other.

2.2.3 **Slice Decomposition and Polygon Descriptors**

A few docking algorithms use slice decompositions of the protein surfaces to identify complementary surface regions. The first such algorithm was published by Walls and Sternberg [112] in 1992. In this section, we will summarize an approach by Helmer-Citterich and coworkers [9, 49], which decomposes the proteins into parallel slices whose boundaries are represented by polygons. The approach determines conformations that exhibit a good match between the polygonal chains of the slices of A and B.

The structure generation procedure, called SHAPES, freezes the conformation of protein A and cuts its solvent accessible surface into parallel slices 1.5 Å thick, orthogonal to the z-axis. Each slice is represented by a 2-D polygon (defined by the intersection of the molecular surface with a plane parallel to the x–y plane) that approximates the protein surface. This is illustrated in Figure 4.

The algorithm considers a discrete set of orientations of protein B that are generated by rotations (R_x, R_y) of B around the x- and y-axes. Each such orientation $(r_x, r_y) \in (R_x, R_y)$ of B is tested in the following way:

(i) For each orientation of protein B, SHAPES calculates the slice decomposition of the solvent accessible surface of protein B, orthogonal to the z-axis, and the respective polygon representations of the slices.

(ii) The algorithm scans a set of possible translations of B along the z-axis. Only translations T_z that map the z-plane of at least one slice polygon of B onto the plane of a slice polygon of A are evaluated using polygon-matching techniques. The algorithm starts with the translation that maps

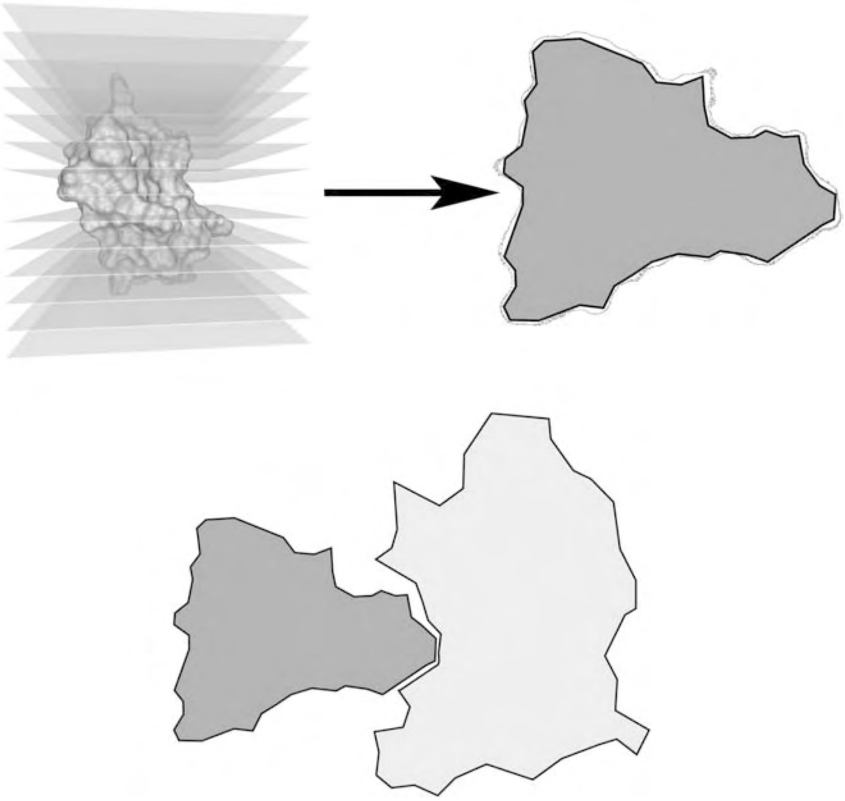


Figure 4 Slice decomposition approaches consider parallel equidistant planes cutting the protein surface. The contour of a surface cut is approximated by a polygon. The algorithm then searches for pairs of compatible polygons of A and B.

the z-plane of the top polygon of B onto the z-plane of the bottom polygon of A. Repeated shifts of 1.5 Å along the z-axis generate the other possible z-translations of B. For each such translation $t_z \in T_z$ of B, the pairs of slice polygons on all different matching z-planes are tested by the polygon-matching technique described below.

(iii) Given a polygon P_A of A and a polygon P_B of B on the same z-plane/slice, the algorithm superimposes each of the m polygon sides of P_B with each of the n polygon sides of P_A by calculating a translation t_x, t_y and a rotation r_z around the z-axis that maps the side of P_B onto the side of P_A. For each such transformation (t_x, t_y, r_z), a complementarity value is calculated.

(iv) The protein–protein interface is a sufficiently large region of complementarity spread across several z-planes. It can thus be found by searching for a set of compatible transformations, both within a single z-plane or across multiple z-planes. This search can be done efficiently by clustering the calculated transformations (t_x, t_y, r_z) of all planes using a 3-D grid with cell size of 3 Å for the translations along the x- and y-axes and 12° for the rotations around the z-axis. The "complementarity values" of all transformations that belong to one cell of the transformation grid are added up. Thus, the cells with the highest complementarity values represent the best partial transformations. The combination of each of these partial transformations (t_x, t_y, r_z) with the current transformation (r_x, r_y, t_z) of B defines a rigid transformation generating a candidate.

The conformation of protein A is fixed during the whole procedure described above. Since protein A is decomposed into slices orthogonal to the z-axis, the poles of A with respect to the z-axis are poorly described. To solve this problem, Helmer-Citterich and coworkers propose to repeat the above procedure with a slice decomposition of A along either the x- or y-axis, orthogonal to the first slice decomposition along the z-axis.

2.2.4 Critical Surface Points and Geometric Hashing

Many docking algorithms determine putative complex conformations by investigating the topological properties of the surfaces of the proteins and by identifying geometrically complementary surface regions. Some of these algorithms apply the following docking strategy: based on their curvature properties, the surfaces are decomposed into patches which are classified as either convex (*knob*), concave (*hole*) or flat (*saddle*). Each surface domain is represented by a central point located on the respective domain or face. These sets of so-called critical points or interest points are sparse representations of the molecular surface. The algorithms determine candidates by searching for matching complementary point sets.

In 1986, Connolly [22] presented the first docking approach matching knobs and holes on the so-called molecular surfaces (solvent excluded surface) [21, 23] of the proteins A and B. Connolly's algorithm generates tentative complex conformations by searching for a rigid transformation that maps four knobs/holes on the surface of A onto four complementary holes/knobs with similar distances on the surface of B.

Wang [113] modified the definition of critical points slightly and matched only one knob/hole on surface A with a complementary hole/knob on the surface of B. The matching of two critical points defines the required translation. Two vectors pointing towards the respective centroids of the local volumes of the hole and the knob are used to fix two rotational degrees of freedom. Furthermore, the approach considers a discrete set of rotations around the axis passing through the centroids to fix the last rotational degree of freedom.

Nussinov and coworkers [38,76] presented a docking algorithm that matches pairs of critical points along with their surface normals. They introduced an elegant and efficient technique for identifying matching pairs of critical points, the geometric hashing technique, developed originally for computer vision applications. The hashing technique is based on a transformation invariant representation of the surface descriptors. Lenhof [73] also applied geometric hashing techniques to generate tentative complex conformations by matching similar triangles of surface points. Geometric hashing is described in more detail in Chapter 16.

2.2.5 Other Approaches

Numerous sophisticated geometric-matching methods have been published in the last two decades. Since a detailed discussion of these methods would go beyond the scope of this work, we will briefly summarize the main ideas behind the matching techniques.

Ackermann and coworkers [2] use geometric and chemical features to decompose the surfaces into domains. An approach based on semantic nets is applied to study the sizes and shapes of the surface domains and to match complementary surface regions. Exner and Brickmann [14,36] compare topographical properties of surface domains using fuzzy logics strategies. Levine and coworkers [74] combine techniques from genetic algorithms, parallel and distributed computing, and virtual reality in the docking program STALK. Gardiner and coworkers [41] propose a genetic algorithm for identifying the areas of greatest surface complementarity. The *soft docking* approach developed by Jiang and Kim [56] integrates dot and cube representation of the molecular surfaces. Cherfils and coworkers [20] calculate putative conformations using a simulated annealing method. In 1992, Bacon and Moult [10] published an approach for protein–protein and protein–ligand docking that uses surface complementarity and electrostatic energy to assess candidate conformations. The algorithm generates candidate conformations by matching patterns of points on the surfaces using McLachlan's [83] least-squares best-fit algorithm.

2.3 Realizing Protein Flexibility

RBD approaches usually succeed in predicting the complex conformations if the bound structures of A and B are sufficiently close to the native, unbound structures. However, even small changes in the conformation of a few side-chains may cause RBD approaches to fail. Although the algorithms will usually still generate approximations of the complex structure, these will be assigned poor scores as they contain physically impossible overlaps between the two proteins. The challenge is now to identify the true positives among the candidates.

The general strategy for realizing protein flexibility is to reconstruct a physically meaningful complex structure. All conformations in the candidate list have to be perturbed and optimized to remove overlaps and to re-establish native contacts in the docking interface. This process requires both an efficient optimization method and a reliable scoring function. As a result of this structural optimization and a subsequent re-scoring, the rank of the true positives will improve.

Studies of the differences between bound and unbound structures indicate that in many cases the protein backbones change only slightly [13, 94]. However, significant conformational changes of side-chains at the docking interface may accompany complex formation. Many docking algorithms thus consider the protein backbones as rigid and simply optimize (demangle) the side-chains of the protein–protein interface.

In contrast to side-chain rearrangements, hinge-bending is the result of conformational changes in the protein backbone. This frequently occurs for multidomain proteins. In these cases, domains of the protein are connected by flexible loop regions. These regions serve as hinges around which the domains may swivel. In some cases, this results in quite complex movements. In the case shown in Figure 6 the two domains of calmodulin can perform a gripper-like movement and close around a possible ligand.

Even more challenging cases of protein flexibility have been described in the literature. In 2000, Shoemaker and coworkers [104] presented a possible answer to the question of why so many proteins in the cell seem to be unfolded most of the time. They proposed a new binding mechanism, the so-called "fly-casting", where binding and folding occur simultaneously. Furthermore, Shoemaker and coworkers argue that a relatively unstructured protein molecule can have a greater capture radius for a specific binding site than the folded state with its restricted conformational freedom. This binding mechanism defines a new grand challenge with respect to the development of structure prediction methods because binding and folding occur more or less simultaneously. While force field-based molecular dynamics methods should in principle be able to handle these cases of full flexibility, these techniques are prohibitively expensive in terms of CPU time.

In the following, we will first discuss approaches for side-chain placement (SCP), then an algorithm for handling hinge-bending by Sandak and coworkers [102], and finally a technique that allows for realizing backbone and side-chain flexibility simultaneously developed by Abagyan and Totrov [1, 109].

2.3.1 Side Chain Placement

The main goal of SCP algorithms is to find the global minimum-energy conformation of the side-chains at the protein–protein interface with respect to a given energy function E. For each given conformation in the candidate list, the placement algorithms determine the side-chains near the contact site and optimize the side-chain conformations. Finally, the optimized conformations are re-ranked with respect to their energy values.

Since side-chain optimization techniques can also be applied to modeling and predicting protein structures, a wide range of methods is available today. Simulated annealing [51, 53, 72], Monte Carlo methods [24, 25, 52, 75, 118], mean-field optimization [62, 71, 84, 85], artificial neural networks [53, 62] and local homology modeling [68] have been applied to this problem.

A large number of techniques for SCP are based on a discretization of conformational space. Ponder and Richards [97] observed that the side-chain conformations occurring in proteins can be adequately described by a rather small, discrete set of rotamers for each residue (for a detailed discussion of rotamers, see also Chapter 10). This discretization results in a combinatorial optimization problem: identify the set of rotamers with the minimal energy, the so-called *global minimum-energy conformation* (GMEC). We will now discuss some of the techniques that have been proposed for identifying the GMEC.

Given a putative complex conformation, SCP methods start by identifying all residues in the protein interface. A residue is considered to be part of the interface, if any of its atoms is within a cutoff distance, e.g. 6.0 Å, from any of the atoms of the other protein.

Let R denote the set of residues that belong to the interface. Note that we do not distinguish between residues of A and B because it is not required for the presentation of the algorithmic techniques. For each residue i, we determine its set R_i of possible rotameric states i_r from a given rotamer library, e.g. the rotamer library of Dunbrack and coworkers [31] (see also Chapter 10). The remaining residues and the backbone of the proteins A and B, which will be kept rigid, are called the template t.

Each possible conformation of the side-chains in R can be described by a binary incidence vector:

$$X = (x_1^1, x_2^1, \cdots, x_r^i, \cdots), \tag{4}$$

where x_r^i equals 1 if residue i is in state r and 0 otherwise. Since "allowed" conformations assign exactly one rotameric state to each residue, the following equations must hold for all possible binary solution vectors X:

$$\sum_r x_r^i = 1 \quad \text{for all residues } i. \tag{5}$$

The GMEC is the combination X of rotamers that fulfills constraint (5) and yields the lowest total energy with respect to a potential energy function

E (note that we restrict ourselves to the typical case of energy functions containing at most pairwise interaction terms):

$$E(X) = E_\mathrm{t} + \sum_{i,r} x_r^i E(i_r) + \sum_{i,r} \sum_{j<i,s} x_r^i x_s^j E(i_r, j_s), \tag{6}$$

where E_t is the potential energy of the template, $E(i_r)$ is the potential energy of side-chain i in state r interacting with the atoms of the template and $E(i_r, j_s)$ is the pairwise potential energy of side-chain i in state r with side-chain j in state s. The GMEC problem is not only NP-complete [96], but also inapproximable [18].

Note that the potential energy of the template E_t is a constant that does not depend on the side-chain rotamers chosen. In a first preprocessing step, the algorithms calculate and store the template energy E_t and the rotamer energies $E(i_r)$. Similarly, all pairwise interaction energies $E(i_r, j_s)$ are calculated.

Typical protein interfaces consist of 40–60 side-chains, yielding up to 10^{60} rotamer combinations. Hence, exhaustive search [16,114,115] is only tractable for very small examples with few side-chains. In the following we will discuss several important techniques for solving the GMEC problem: the *dead-end elimination* (DEE) technique introduced by Desmet and coworkers [27], an A^* algorithm developed by Leach and Lemon [69, 70], and integer linear programming (ILP) techniques.

2.3.1.1 **Dead End Elimination** The DEE method developed by Desmet and coworkers [27] reduces the size of the search space by eliminating side-chain rotamers that are incompatible with the GMEC. A rotamer i_r can be safely ignored in the search for the GMEC, if a second rotamer i_u of side-chain i exists such that the following inequality is fulfilled:

$$E(i_r) + \sum_{j \neq i} \min_s E(i_r, j_s) > E(i_u) + \sum_{j \neq i} \max_s E(i_u, j_s). \tag{7}$$

This so-called DEE theorem states that the rotamer i_r cannot be part of the GMEC if its "best" (i.e. lowest) energy is still larger than the "worst" (i.e. highest) energy of i_u. Desmet and coworkers proved an analogous theorem for rotamer pairs and successively applied the two theorems in an iterative fashion as long as the number of rotamers could be reduced. Applying DEE will usually not result in a single solution for the GMEC, but it drastically reduces the search space. The remaining search space can then be more effectively searched using one of the techniques described below.
A more effective variant of the DEE theorem has been formulated by Goldstein [44]: a rotamer i_r cannot be part of the GMEC, if a second rotamer i_u of side-chain i exists such that the following inequality holds:

$$E(i_r) - E(i_u) + \sum_{j \neq i} \min_s \{E(i_r, j_s) - E(i_u, j_s)\} > 0. \tag{8}$$

In the last decade, a series of papers on variants of the DEE theorem has been published [28, 29, 44–46, 67, 78]. Desmet and coworkers [29] discuss further variants of the DEE theorem and summarize algorithms for implementing these inequalities efficiently.

2.3.1.2 **"Branch & Bound" and the A^* Algorithm** In principle, all possible solutions to the GMEC can be represented by an enumeration tree. Each path from the root to a leaf of the tree describes a feasible rotamer combination. The i-th layer of the tree represents the rotamers of the i-th side-chain and each node is labeled with a rotameric state i_r of i. Hence, the number of layers equals the number of side-chains (Figure 5). A path $P(v)$ from the root to a node v in layer m visits exactly one node in each layer. If the node in layer i is labeled i_r, then $P(v)$ assigns rotameric state i_r to residue i. Hence, $P(v)$ fixes the rotameric states of the first m side-chains. The subtree $T(v)$ of node v represents all possible rotamer combinations of the remaining side-chains.
During the construction of the tree, the potential energies of the different paths are calculated and stored in the tree nodes where the paths end. Hence, the root of the tree stores the template energy E_t while the other nodes v store the total potential energy of all rotamers that lie on the path $P(v)$ from v to the root:

$$E(v) = E_\mathrm{t} + \sum_{i_r \in P(v)} E(i_r) + \sum_{i_r \in P(v)} \sum_{\substack{j_s \in P(v) \\ j<i}} E(i_r, j_s) \tag{9}$$

The GMEC can then be determined by constructing the whole tree and by searching for the leaf with the lowest energy. The path from this leaf to the root describes the rotamer combination with minimal potential energy. Since the number of rotamer combinations grows exponentially with the number of side-chains, building the whole enumeration tree is not feasible for typical

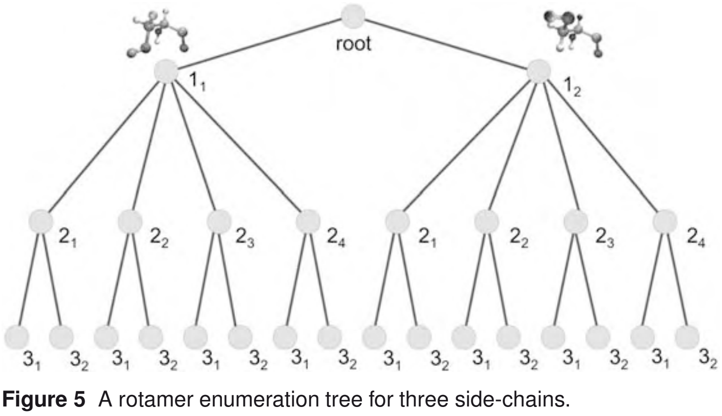


Figure 5 A rotamer enumeration tree for three side-chains.

problem sizes. "Branch & bound" approaches are widely used methods for solving high-dimensional optimization problems. These algorithms try to avoid the exploration of subtrees that cannot contain a minimal conformation.

Let U be an upper bound for the energy value of the GMEC and $E_{\mathrm{L}}(v)$ a lower bound for the energetic contribution of the best path from a node v to any of the leaves in the subtree $T(v)$ of v, then the subtree $T(v)$ can be safely ignored if the following inequality holds:

$$E(v) + E_{\mathrm{L}}(v) > U \tag{10}$$

The upper bound U can be obtained using a greedy or multi-greedy heuristic (see, e.g. Refs. [89, 98]) that usually generate good approximations of the optimal solution. A reasonable lower bound $E_{\mathrm{L}}(v)$ for the potential energy of the optimal path in $T(v)$ can be calculated as follows. We have to add up lower bounds $L(j)$ for the energetic contribution of the side-chains j that have not yet been assigned a rotameric state, i.e. the side-chains not represented by the nodes in $P(v)$:

$$E_{\mathrm{L}}(v) = \sum_{j \notin P(v)} L(j) \tag{11}$$

Suitable lower bounds $L(j)$ can be determined as follows:

$$L(j) = \min_{s} \left\{ E(j_s) + \sum_{i_r \in P(v)} E(j_s, i_r) + \sum_{k \notin (P(v) \cup j)} \min_{l} E(j_s, k_l) \right\} \tag{12}$$

The first sum in the above definition describes the pairwise interaction energies of rotamer j_s with the side-chains on the path $P(v)$ that have already been assigned a rotameric state, whereas the last sum in Eq. (12) is a lower bound for the pairwise interaction energies of j_s with the other side-chains that have not yet been assigned a rotameric state.

"Branch & bound" approaches usually build the tree layer by layer in a breadth-first search manner. Before adding the children of node v and starting the construction of the subtree of node v, the algorithm calculates $E_{\mathrm{L}}(v)$ and applies inequality (10) to test if the subtree $T(v)$ can be safely ignored.

An alternative to "Branch & bound" is the A^* algorithm introduced by Hart and coworkers [47]. It has been applied to SCP for protein–ligand docking by Leach and Lemon [69, 70]. The A^* algorithm is a heuristic procedure for searching an optimal path in a graph or tree that allows to explore deeper layers of the tree before completing upper layers, if promising paths have been detected. The algorithm first visits the children v of the root, and calculates the potential energies $E(v)$ and the lower bounds $E_{\mathrm{L}}(v)$ for the best possible path in the corresponding subtrees $T(v)$. For each child v of the root, the sum of

$E(v)$ and $E_{\mathrm{L}}(v)$ serves as an estimator:

$$BP(v) := E(v) + E_{\mathrm{L}}(v) \tag{13}$$

for the best possible energy. These estimators are stored in a priority queue together with their nodes such that the first element v is the most promising node with the smallest estimator $BP(v)$.

In the next step, the A^* algorithm further explores the tree by adding new nodes representing the children w of the node v with the best estimator $BP(v)$. Then, the estimator $BP(v)$ is removed and the estimators $BP(w)$ of the new nodes are computed and added to the priority queue. The algorithm stops if an optimal solution has been found. An optimal solution is one that assigns a rotameric state to each of the side-chains and has an energy that is lower than the best estimator stored.

2.3.1.3 **Integer Linear Programming** ILP techniques have been very successfully applied to many high-dimensional optimization problems. We will now derive an ILP formulation for the GMEC problem.

If we combine Eqs. (5) and (6) for the GMEC problem, we arrive at:

$$\begin{aligned}
\min_X E(X) \quad &= \quad \min_X \left\{ E_{\mathrm{t}} + \sum_{i,r} x_r^i E(i_r) + \sum_{i,r} \sum_{j<i,s} x_r^i x_s^j E(i_r, j_s) \right\} && (14)\\
\text{s.t.} \quad & \quad \sum_r x_r^i = 1 \quad \text{for all } i \in R && (15)\\
& \quad x_r^i \in \{0,1\} \quad \text{for all } i \in R \text{ and } r \in R_i && (16)
\end{aligned}$$

This is an integer quadratic programming problem, since the solutions are integer and since the energy function contains products of the form $x_r^i x_s^j$. It can be transformed into an ILP problem using the following procedure:

(i) We determine the largest pairwise interaction energy $E_{\max}$ of any pair of rotamers.

(ii) We substitute the pairwise interaction energy $E(i_r, j_s)$ of each rotamer pair by $E(i_r, j_s) - (E_{\max} + \varepsilon)$ where ε is a small positive constant.

(iii) For each pair of variables $x_r^i x_s^j$, we introduce a new binary variable y_{rs}^{ij} that takes the value 1 if and only if x_r^i and x_s^j are 1.

The first two steps are necessary to ensure that all pairwise interaction energies are negative. The above transformation shifts the solutions of the GMEC problem only by a constant. Hence, the minima of the transformed problem are identical to the minima of the untransformed problem:

$$\begin{aligned}
\min_X E(X) &= \min_X \left\{ E_\mathrm{t} + \sum_{i,r} x_r^i E(i_r) + \sum_{i,r} \sum_{j<i,s} y_{rs}^{ij} E(i_r, j_s) \right\} && (17)\\
\text{s.t.} \quad & \sum_r x_r^i = 1 \qquad \text{for all } i \in R && (18)\\
& y_{rs}^{ij} \le x_r^i \qquad \text{for all } i, j \ne i \in R, r \in R_i, s \in R_j && (19)\\
& y_{rs}^{ij} \le x_s^j \qquad \text{for all } i, j \ne i \in R, r \in R_i, s \in R_j && (20)\\
& x_r^i \in \{0,1\} \qquad \text{for all } i \in R, r \in R_i && (21)\\
& y_{rs}^{ij} \in \{0,1\} \qquad \text{for all } i, j \ne i \in R, r \in R_i, s \in R_j && (22)
\end{aligned}$$

The linear constraints (19) and (20) guarantee that the variable y_{rs}^{ij} can take the value 1 only if x_r^i and x_s^j are 1. If, in addition, $x_r^i = x_s^j = 1$, then the fact that $E(i_r, j_s)$ is negative forces the variable y_{rs}^{ij} to take the value 1 because we are dealing with a minimization problem. In other words, if $y_{rs}^{ij} = 0$ held for a minimal conformation, setting $y_{rs}^{ij} = 1$ would not violate any of the above constraints. However, it would generate a new solution whose energy is smaller than the energy of the supposed minimal conformation by a factor of $E(i_r, j_s)$, thus contradicting the above assumption.

Althaus and coworkers [5, 6] presented the first ILP formulation of the GMEC problem and a "Branch & cut" algorithm. "Branch & cut" is the most common technique for solving ILP problems. For details on "Branch & cut" and integer linear programming, we refer to the book by Wolsey [117]. The scheme of standard "Branch & cut" approaches can be summarized as follows. We relax the ILP by dropping the integer condition – we allow the variables x_r^i to take any real value in the interval $[0,1]$ – and solve the resulting linear program (LP). If the solution $\bar{X}$ is integral we have found the GMEC. Otherwise, we search for a valid linear inequality $f(X) \le f_0$ that cuts off the solution $\bar{X}$, i.e. $f(X) \le f_0$ for all feasible solutions X of the ILP problem and $f(\bar{X}) > f_0$. The set $\{X | f(X) = f_0\}$ is called a cutting plane. Any cutting plane found is added to the LP and the resulting LP is solved again. The generation of cutting planes is repeated until either an optimal solution is found or the search for a cutting plane fails. In the second case a branch step follows, i.e. we generate two subproblems by setting one fractional variable x_r^i of the last solution to 0 in the first subproblem and to 1 in the second subproblem. We then start the "Branch & cut" algorithm for the two subproblems recursively. This recursion gives rise to an enumeration tree of subproblems, where the minimum of the solutions of the two subtrees represents the optimal solution of our primal ILP problem.

The search for cutting planes is called the separation problem. The most promising cutting planes are the facets of the convex hull of all feasible solutions. In order to find good cutting planes, Althaus and coworkers [5] studied the convex hull of the feasible solutions, the so-called GMEC polyhedron, and identified a few classes of facet-defining inequalities.

Other ILP formulations of the GMEC problem have been presented by Eriksson and coworkers [34] and by Kingsford and coworkers [59]. The compact graph-theoretic ILP formulation of Kingsford and coworkers can be summarized as follows:

Let G be an undirected $|R|$-partite graph with node set:

$$V := V_1 \cup V_2 \cup \cdots \cup V_{|R|}, \tag{23}$$

where V_i contains a node u for each rotamer i_r of side-chain i. The potential energy $E_{uu} := E(i_r)$ is assigned to the node $u \in V_i$ that represents the rotamer i_r. Each pair $(u, v) = (i_r, j_s)$ of rotamers with $i \neq j$ is connected by an edge with weight $E_{uv} = E(i_r, j_s)$. A feasible solution of the GMEC problem has to pick exactly one node per node set V_i. The energy of the feasible solution is the sum of the template energy E_t, the node weights and the weights of the edges that connect the selected nodes. We introduce binary decision variables x_{uu} and x_{uv} for each node $u \in V$ and for each edge (u, v) of the graph:

$$\begin{aligned}
\min_X E(X) &= \min_X \left\{ E_\mathrm{t} + \sum_{u \in V} x_{uu} E_{uu} + \sum_{(u,v) \in G} x_{uv} E_{uv} \right\} && (24)\\
\text{s.t.} \quad & \sum_{u \in V_i} x_{uu} = 1 \qquad \text{for all } i \in R && (25)\\
& \sum_{u \in V_i} x_{uv} = x_{vv} \qquad \text{for all } i \in R \text{ and } v \notin V_i && (26)\\
& x_{uu}, x_{uv} \in \{0, 1\} \quad \text{for all } u, v \in V. && (27)
\end{aligned}$$

The constraints (26) ensure that exactly those edge weights E_{uv} are added to the energy function for which both end nodes u and v are selected, i.e. $x_{uu} = 1$ and $x_{vv} = 1$.

Kingsford and coworkers [59] show that their ILP formulation allows for solving large problem instances. They propose to relax the ILP to an LP. If the solution of the LP is integral, the SCP problem has been solved in polynomial time. Otherwise, they start the computationally more expensive ILP procedure. Surprisingly, they show that optimal solutions can almost always be obtained directly by solving the respective LP when placing side-chains on native or homologous backbones, whereas the design problem often cannot be solved using LP. These results have been obtained with an energy function based on van der Waals interactions and a statistical rotamer self-energy term. Kingsford and coworkers also found that small modifications of

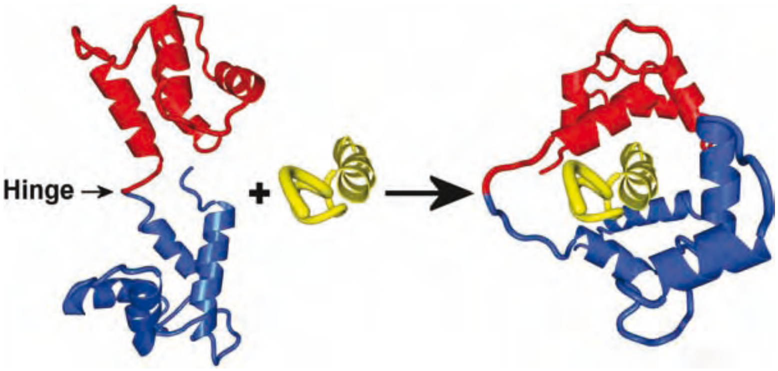


Figure 6 Hinge-bending accompanies the binding of calmodulin and to a peptide. The two terminal domains of calmodulin "wrap" around the peptide by a movement around the hinge region marked by the arrow (PDB IDs 1CFF and 1BBM).

the energy function may lead to an increase of examples for which the more expensive ILP procedure is required.

2.3.2 **Hinge-bending**

Sandak and coworkers [101, 102] published the first algorithm for handling hinge-bending. To accurately predict the structure of protein–protein and protein–ligand complexes, they adapted a technique developed in computer vision and robotics for the efficient recognition of partially articulated objects. These objects consist of rigid parts connected by rotary joints. The approach is based on an extension and generalisation of the Hough transform and the geometric hashing paradigm for rigid objects.

Since a detailed description of this sophisticated algorithm would go beyond the scope of this work, we focus on the core ideas of the method. To simplify matters, we assume that

(i) The ligand B is built of two "rigid" domains B_1 and B_2 connected by a short flexible hinge.

(ii) The position of the hinge in protein B is known and can be described by a point H.

(iii) Apart from the six degrees of motional freedom of B as a whole, independent rotations of B_1 and B_2 around the hinge are allowed as additional conformational changes.

The algorithm starts by applying a slightly modified version of the RBD scheme based on geometric hashing as described in Refs. [38, 76] (see also

Chapter 16). Triplets of points on the surfaces of A and B are used to calculate all rigid transformations T of B that result in interesting contacts between the surfaces of A and B. If a transformation t is generated by a surface matching between B_1 and A, t is associated with the domain B_1 and with B_2 otherwise.

The hinge-bending algorithm is based on the following concept (see Figure 7): hopefully, the set T of possible transformations contains a transformation t_1 placing B_1 close to the correct position, but not necessarily B_2, and a transformation t_2 placing B_2 correctly, but not necessarily B_1. Both transformations share a common feature: when applied to their respective domains, they will place the hinge at the same position. The algorithm thus tries to identify pairs of transformations associated with both domains that would place the hinge at approximately the same position using a grid-based technique.

Sandak and coworkers apply all transformations in T to H resulting in a set $T(H)$ of possible locations of the hinge. These putative hinge locations are clustered using a 3-D grid. For each grid cell, a vote accumulator counts the number of transformations that map H into this cell. The algorithm considers a cell as interesting if it receives a large number of votes from transformations associated with both domains. The high-scoring cells and the associated transformations are selected, while the other transformations are removed. Each of the remaining transformations is applied to its domain and checked for collisions with A. Transformations that yield large overlaps are removed from the candidate list, similarly those yielding an insufficient number of

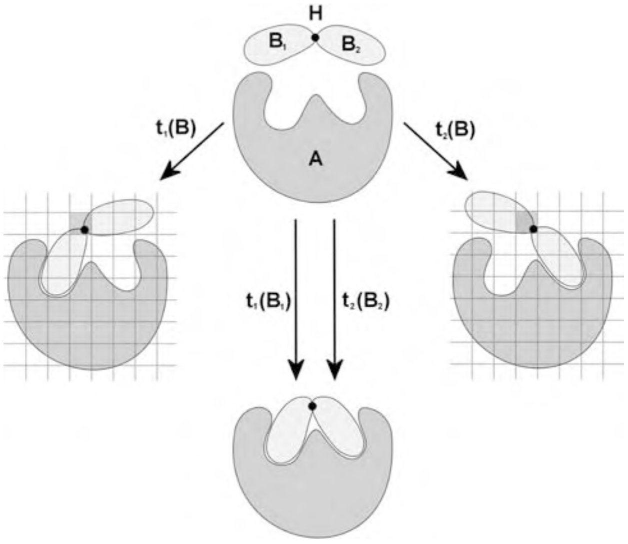


Figure 7 Schematic illustration of protein-docking with hinge-bending: the two domains B_1 and B_2 have to swivel around their joint hinge H to fit into the binding site of A.

favorable van der Waals contacts with A. In a following self-collision test, the remaining possible pairs (t_1, t_2) of transformations belonging to high-scoring cells where t_1 is associated with B_1 and t_2 with B_2 are applied to their respective domains. If the transformed domains $t_1(B_1)$ and $t_2(B_2)$ do not show a forbidden self-collision, the pair of transformations (t_1, t_2) is added to the set of possible solutions together with a geometric score, the sum of the van der Waals contacts of $t_1(B_1)$ with A and $t_2(B_2)$ with A.

2.3.3 **Biased Probability Monte Carlo (BPMC) Conformational Search**

Abagyan and Totrov [1, 109] suggested an efficient and effective method for optimizing the backbone conformation and the side-chain positions simultaneously. The modified Monte Carlo method, called the BPMC approach, operates in torsion angle space. To simplify matters, we will sketch the BPMC algorithm using structure prediction of peptides as an application example.

In torsion angle space, the conformation of a peptide can be described by the backbone torsion angles ϕ, ψ and the torsion angles $\chi_\alpha, \chi_\beta, \cdots$ of the side-chains. Abagyan and Totrov combine the two torsion angles ϕ, ψ of each residue to one "subspace" or internal variable of the optimization problem such that the number of internal variables representing the backbone is equal to the number of residues. The torsion angles of individual residues can also be combined to one subspace or internal variable.

The BPMC approach is based on the idea "to sample with larger probability those regions of the conformational space which we know *a priori* are, on the average, highly populated and to sample with less probability regions known to be less populated" [1]. Abagyan and Totrov prove that random steps chosen according to the expected probability distribution of the "subspace" parameters should be preferred over any other way to make a random move. Therefore, they carry out a statistical analysis of the local conformational preferences of a representative set of protein structures to identify preferred zones in ϕ–ψ and χ subspaces. The calculated ϕ–ψ and χ maps are divided into regions by visual inspection or according to the maxima of the torsion potential. Each region defines a preferred zone z that is approximately described by an ellipse. These zones are also associated with probabililties $P(z)$, which are derived from the number of points contained in the zones.

The BPMC iteration scheme can be summarized as follows:

(i) The BPMC procedure randomly selects a subspace or internal variable, i.e. it chooses a backbone or side-chain torsion angle belonging to one residue and the respective subspace.

(ii) It randomly selects one zone z_k according to the probability $P(z_k)$ out of the set of all high-probability zones $z_1, z_2, \cdots, z_m$ of the selected internal variable or subspace.

(iii) It makes a normally distributed step in the vicinity of the z_k-th zone.

(iv) A local minimization of the new conformation is carried out using the ECEPP/2 energy function [37, 92].

(v) The total energy of the optimized conformation is calculated by adding additional energy terms for solvation energy, electrostatic free energy and entropy.

(vi) The conformation is accepted if the Metropolis criterion [86], which is the normal acceptance criterion for a move in the simulated-annealing procedure, is fulfilled.

The BPMC approach has been successfully applied to a number of different optimization and structure prediction problems, ranging from *ab initio* prediction of peptide structures to protein–protein docking.

2.4 Scoring Functions

To distinguish the native structure from decoys, the generated candidates are usually evaluated using some kind of scoring function. The number of different scoring functions that have been used in protein–protein docking is very large. Therefore, we will discuss some general requirements for these functions first and then describe selected scoring functions in more detail. (For a discussion of scoring functions in the context of protein–ligand docking, please refer to Chapter 16.)

The ideal scoring function is (i) highly accurate, (ii) computationally efficient and (iii) robust with respect to small deviations from the native structure. Such an ideal function would allow us to easily compute the interaction free energies of all candidates and then pick the one with the lowest score/energy, which is hopefully close to the native structure.

Unfortunately, the prediction of binding free energies is a challenging problem. While the biophysical effects contributing to the binding free energy ΔG_{bind} are well known, they are not yet understood to the extent necessary to construct an accurate scoring function. It is thus difficult to model these effects adequately. This is particularly true for entropic contributions and solvation effects. However, even for the well-understood contributions like hydrogen bonds or van der Waals interactions it is hard to find sufficiently accurate, albeit computationally efficient, models.

Over the years, different classes of scoring functions have been developed, each with its own strengths and weaknesses. All these approaches share one major simplification: while the thermodynamic properties we try to predict (free energies in particular) are defined as ensemble properties, all scoring functions try to deduce this property based on a single structure only. There has been some effort to predict binding free energies for small molecule ligands using ensemble techniques (free energy perturbation and thermodynamic integration), however these methods are not applicable to large protein–protein complexes.

Most scoring functions fall into one of three categories: (i) *ab initio* models, (ii) empirical models and (iii) knowledge-based models. The first category is based on first principles, i.e. on solid physical theories like quantum mechanics. These models require no additional parameterization besides physical constants. While they are theoretically elegant, they are often computationally intractable for the large systems considered in protein–protein docking. Current scoring functions thus mostly fall into the latter two categories (empirical and knowledge-based), which we will discuss below. A comprehensive review of the different types of scoring functions in the context of protein–ligand docking can be found in [15].

2.4.1 Empirical Potentials

Empirical potentials decompose the overall binding free energy into individual independent contributions, which are then approximated by separate models. Many simple scoring functions described for RBD approaches fall into this category. Empirical models are often, but not always, based on solid theoretical models. In particular, some of these potentials are derived from or inspired by molecular mechanics force fields (AMBER or CHARMM in most cases). These force fields are also empirical scoring functions in a broader sense and have been used for scoring in protein–protein docking as well.

Empirical scoring functions use varying decompositions of the total binding free energy into individual contributions. The decompositions aim at being sufficiently *complete*, *accurate* and *orthogonal*. Sufficient completeness implies that all major effects governing protein association are covered. If this is not the case, then sufficient accuracy cannot be achieved. Orthogonality implies that each effect is covered by one contribution of the decomposition only, thus simplifying the validation of the scoring function and yielding a minimal and concise decomposition.

Obviously, there are numerous possible contributions to the total binding free energy. A reasonably complete decomposition of the binding free energy ΔG_{bind} could, for example, include the following contributions:

$$\begin{aligned}\Delta G_{\text{bind}} = \Delta G_{\text{vdW}} + \Delta G_{\text{es}} + \Delta G_{\text{hp}} + \Delta G_{\text{hb}} \\ + \Delta G_{\text{solv}} + \Delta G_{\text{int}} + \Delta G_{conf} - T\Delta S_{\text{tr/rot}}\end{aligned} \tag{28}$$

where ΔG_{vdW} are the van der Waals contributions, ΔG_{es} is the change in electrostatic free energy, ΔG_{hp} accounts for the hydrophobic interaction, ΔG_{hb} accounts for the hydrogen bond energy, ΔG_{solv} accounts for the change in solvation free energy, ΔG_{int} accounts for the change in the internal energy of the two proteins (e.g. conformational changes), ΔG_{conf} accounts for the loss of side-chain torsional degrees of freedom, and $T\Delta S_{\text{tr/rot}}$ accounts for the loss of translational and rotational degrees of freedom upon association. Clearly, this decomposition is neither entirely orthogonal nor entirely complete. Hydrogen bonds and solvation effects also have an electrostatic contribution, and the hydrophobic effect is not completely separable from van der Waals and solvation effects. Nevertheless, this decomposition accounts for the major interactions occuring between proteins. The art of constructing an empirical scoring function lies in the choice of good approximations for each of these effects and in knowing which of these contributions can be neglected for the problem at hand.

In the simplest case, an empirical scoring function accounts for a single effect only, e.g., for hydrophobic interactions or electrostatic contributions. Surface overlaps used as score in correlation-based approaches can be regarded as simple empirical scoring functions accounting for the hydrophobic effect and van der Waals interactions only. While these trivial scoring functions will work readily for certain types of protein–protein interactions (e.g. large hydrophobic interfaces for correlation approaches), they will fail as soon as the total binding free energy is dominated by other effects. Sophisticated scoring functions thus aim at modeling as many of the effects as accurately as possible.

The scoring function used by Jackson and Sternberg [54] in their RBD approach is an example of a more sophisticated empirical scoring function. The approach relies on a subset of the above decomposition:

$$\Delta G_{\text{bind}} = \Delta G_{\text{es}} + \Delta G_{\text{solv}} + \Delta G_{\text{hp}}. \tag{29}$$

Solvation and electrostatics are handled by the same approach – continuum electrostatics. This approach models the solutes (the proteins) as regions of low dielectric constant immersed in a continuum of high dielectric constant – the solvent (water). The overall electrostatics of this system can be readily solved using the Poisson or Poisson–Boltzmann equation by standard numerical techniques (e.g. finite difference methods, e.g. Ref. [43]). Solving the (linearized) Poisson–Boltzmann equation on a sufficiently fine 3-D grid yields the electrostatic potential for each grid point. Using a quite elaborate scheme, Jackson and Sternberg compute the change in solvation free energies of the two proteins upon association ($\Delta\Delta G^{A}_{\text{solv}}$ and $\Delta\Delta G^{B}_{\text{solv}}$) and their electrostatic interaction energy ($\Delta G^{AB}_{\text{es}}$) from these potentials.

The hydrophobic effect is estimated as the cavitation free energy, i.e. the energy required to form a protein-shaped cavity in the surrounding water. This energy is thought to be roughly proportional to the molecular surface area. Binding of the two proteins obviously reduces surface area, as the protein binding interface is no longer exposed to the solvent. This results in a negative cavitation-free energy. The hydrophobic effect is thus estimated as:

$$\Delta G_{\text{hp}} = \gamma \Delta A = \gamma \quad (A_{AB} - A_A - A_B), \tag{30}$$

where A_{AB}, A_A and A_B are the molecular surface areas of the complex and the proteins, respectively, and γ is a constant.

The full model thus looks as follows:

$$\Delta G_{\text{bind}} = \Delta\Delta G^{A}_{\text{solv}} + \Delta\Delta G^{B}_{\text{solv}} + \Delta G^{AB}_{\text{es}} + \gamma(A_{AB} - A_A - A_B). \tag{31}$$

While the model turned out to be successful for scoring protein–protein complexes, the use of the finite-difference Poisson–Boltzmann equation results in quite long running times. This scoring function is thus viable for the final re-ranking stage of the docking algorithm only.

It should be remarked that the energies predicted by most empirical scoring functions are not able to reproduce the absolute value of the binding free energy with reasonable accuracy. The reason is often to be found in the nonorthogonality of the overall decomposition, in the insufficient accuracy of the individual contributions, but also in the overall balance of these contributions. However, this does not impede the usefulness of these scoring functions for protein–protein docking, where a proper separation of the native structure from the decoys (i.e. a correct relative *ranking* of the energies) is sufficient.

2.4.2 **Knowledge-based Potentials**

In contrast to empirical scoring functions, knowledge-based scoring functions do not explicitly model individual contributions to the binding free energy. Instead, they are based on the frequencies of residue–residue or atom–atom contacts observed in X-ray structures of protein–protein complexes. Those interactions occuring frequently are counted as energetically favorable, whereas interactions observed rarely in complexes are considered less favorable. The frequencies observed can be converted to pseudo-energies assuming a Boltzmann distribution (the so-called *inverse Boltzmann law* [107]). In this case, the potentials are often called *potentials of mean force* (PMFs).

Knowledge-based potentials can be constructed for different model resolutions. The potentials initially developed for protein structure prediction are typically residue-based potentials, i.e. two amino acids are considered to be in contact and thus to contribute to the total energy, if they are within a given threshold distance. In its simplest form, the computation of the interaction free energy reduces to a summation over all pairs in contact across the protein interface:

$$\begin{aligned} &\Delta G_{\text{bind}} = \sum_{(i,j)} e_{t(i)t(j)} \quad \text{for all} \quad i \in A, j \in B \\ &\text{with} \quad r_{ij} := |\vec{r}_i - \vec{r}_j| \leq r_{\text{cutoff}}, \end{aligned} \tag{32}$$

where $t(i)$ denotes the atom/residue type of i, $e_{t(i)t(j)}$ are the statistically derived interaction energies, $\vec{r}_i$ and $\vec{r}_j$ are the positions of i and j, and r_{cutoff} is the cutoff distance, i.e. the maximum distance for which the pair (i, j) is still considered to be in contact (usually $6 - 7$ Å). Less coarse scoring functions compute these $e_{t(i)t(j)}$ as a function of the distance. The above equation then becomes:

$$\Delta G_{\text{bind}} = \sum_{(i,j)} e_{t(i)t(j)}(r_{ij}). \tag{33}$$

Deriving these $e_{t(i)t(j)}$ from a structural database typically involves counting of the individual pairs. The different knowledge-based potentials differ in the way this counting is done (e.g. the assignment of atom/residue types) and, more importantly, they differ in the choice of a reference state. The issue of reference states is beyond the scope of this paper. A recent discussion of the problem can be found in Refs. [77, 122].

The key step in all these approaches is relating the e_{ab} to the number of observed contacts between atoms/residues of type a and b in the database for a given distance. The inverse Boltzmann law yields:

$$e_{ab}(r) = -k_B T \ln \frac{g_{ab}(r)}{g(r)}, \tag{34}$$

where $g_{ab}(r)$ is the probability of finding a pair of types a and b in distance r, and $g(r)$ is some normalized reference probability. These probabilities are approximated by counting contacts in the structural database.

Both, atom contact potentials (ACPs) and residue contact potentials (RCPs) have been used extensively in protein–protein docking with significant success. In theory, knowledge-based potentials account for all effects contributing to the binding affinity; however, some of these effects tend to be underestimated. In particular, the repulsive part of the van der Waals contribution and solvation effects are often added to yield a more reliable scoring function.

The *atomic contact energy* (ACE) proposed by Zhang and coworkers [121] is one of these potentials. It is based on a classification into 18 distinct atom types. For each of these types, the number of contacts was determined from a database of 89 homology-reduced protein structures with a single cutoff radius of 6 Å. Interaction scores can thus be readily computed using Eq. (32). ACE is typically augmented with additional Coulomb electrostatics and entropic contributions to predict protein binding free energies [119].

An example for a residue-based contact potential is RPScore, developed by Moont and coworkers [90]. RPScore considers two residues to be in contact if their C_β atoms are within 6 Å of each other. Counting these contacts across the interface of a set of protein complexes and applying the inverse Boltzmann law to these counts, normalized with the expected number of contacts, results in relative preferences for amino acid contacts. Moont and coworkers used this type of potential and similar atom-based potentials for rescoring docking candidates obtained through correlation docking and yielded very good results.

Murphy and coworkers [91] used RCPs (RPScore) and ACPs (ACE) in subsequent filtering and scoring steps. They found that both types of potentials are good for discriminating native structures from decoys. The performance increases if both methods are combined, i.e. if RPScore is used for an initial filtering and ACE for a final scoring or *vice versa*. In order to account for overlaps, both potentials had to be augmented by an additional van der Waals energy term.

2.5 Data-driven Docking

Today, protein complexes are studied with a wide variety of experimental techniques. Not all of these experiments yield enough information to allow a direct structure elucidation, but nonetheless they often allow to draw useful conclusions about the properties of the complex. Indeed, integrating such experimental data into protein-docking procedures has been consistently reported to improve prediction accuracy. This approach of integrating experimental results into the docking process is commonly known as *constrained* or *data-driven* docking.

A large number of different experimental techniques has been used in the context of data-driven docking. Here, we will focus on select approaches and refer the interested reader to the recent review of the field by van Dijk and coworkers [111]. At this point we would like to emphasize that for many data-driven docking approaches, the experimental origin of the constraints is of no importance. If the docking algorithm is able to use "generic" distance or angle constraints, it can utilize information from the most diverse experimental techniques in a unified framework.

2.5.1 Experimental Techniques

Site-directed mutagenesis of individual amino acids in the complex provides information on the involvement of the mutated residue in the binding interface. If a certain point mutation influences the binding in any way, the residue is considered part of the complex interface. If, on the other hand, replacing a residue does not influence the binding, this can at least be taken as a hint that it is not part of the interface. This information can be exploited in data-driven docking. In the structure generation stage, e.g., we restrict the search to those candidates where the experimentally identified residues are indeed part of the binding interface.

Mass Spectrometry (MS) techniques have become a popular tool for many biological applications, particularly in proteomics. Even though it might not be obvious at first glance, MS experiments can also be used to provide constraints for the protein-docking problem. This can be achieved in two different ways. The first technique is the measurement of H/D exchange rates, where the proteins are immersed in heavy water (D_2O). In this situation, hydrogen atoms on the surface of the protein can be exchanged by deuterium atoms from the heavy water, perturbing the mass of the protein. Measuring these mass differences yields information about the exchange rates. Residues in the protein–protein interface are shielded from H/D exchange and thus experience lower exchange rates in the complex. A second type of constraint can be derived from MS experiments on cross-linked proteins. A reactive bifunctional cross-linker covalently binds to two residues. The chemistry of the functional groups determines at which sites the cross-linking occurs. The protein complexes are then fragmented (e.g. through tryptic digest) and the resulting peptides are investigated using MS techniques. If this reveals a fragment with a mass equal to the sum of the masses of two peptides occuring in the complex plus the mass of the cross-linker, one can derive a simple distance constraint: in the complex, the two peptides occuring in the cross-linked fragment cannot be farther away than the distance of the two functional groups of the cross-linker. Thus, cross-linked pairs of peptides from A and B yield an upper bound on the distance of pairs of residues in the complex.

NMR spectroscopy measures the electromagnetic absorption properties of nuclear spins subjected to a strong magnetic field. These effects sensitively depend on the chemical environment of the absorbing atom and can be used to derive distance and orientational constraints between atom pairs in the complex. The most commonly used constraints are distance constraints based on nuclear Overhauser enhancements (NOE) and orientational constraints based on residual dipolar coupling (RDC). In addition, NMR can be used to determine which residues undergo conformational changes during complex formation by H/D exchange measurements, cross-saturation transfer or chemical shift perturbation. Using NMR to derive distance or angle constraints has become a popular technique for protein docking. Unfortunately, most techniques require time-consuming manual annotation of the NMR spectra (resonance assignment).

2.5.2 **Algorithmic Approaches**

In principle, experimental information can be introduced into the docking process at three different stages: the structure generation stage, the filtering stage or the scoring stage. If experimental data is used in the filtering stage, it is often used in the form of geometric constraints. Candidate structures violating too many of these constraints are removed and the search space is thus reduced. In the scoring stage, the candidate structures can be used to predict an experimental property. The similarity between predicted and experimental data can then serve as a scoring function. Since most of the algorithmic techniques are very similar to the techniques already described above, we will only discuss a few select examples.

In general, the structure generation itself is driven by some kind of energy or scoring function. To integrate experimental data, additional scoring terms can be introduced to penalize deviation from the measured values. A typical example for such an approach is implemented in the docking program HADDOCK [30], which integrates information about interfacial residues by adding distance penalties. Since not all interface residues may be detected, neighboring residues on the protein surface will be added to those validated experimentally. Given a candidate structure, effective distances are calculated for each interface residues. The effective distance measures a residue's proximity to the interface. Candidates where the interface largely coincides with the experimentally determined interface will have low effective distances, resulting in good scores.

Experimental information can also be used for correlation-based docking techniques by weighting different parts of the protein differently, according to the experimental data at hand [11, 12]. For example, the contribution of residues known to be part of the binding site can be increased by assigning their surface regions a larger value. Anand and coworkers [7] used MS H/D data to filter candidates generated with the docking program DOT [81]. NMR data was used as a post-scan filter [95] in the correlation-based program FTDOCK [40].

Integrating experimental data in the scoring phase can be achieved in numerous ways. Kohlbacher and coworkers [63] used unassigned ^{1}H-NMR spectra of complexes to re-score the candidates. In an initial phase, they generate candidates by RBD based on geometric hashing [73]. With an empirical model, they predict the chemical shift of every proton in the candidate structure. The sum of these shifts yields a theoretical spectrum for each candidate. These spectra are then compared to the NMR spectrum of the native structure and the resulting similarities are used to score the candidates. Based on this score alone they achieved very good rankings of several protein–protein complexes.

2.6 Assessment of Docking Predictions

At first glance, assessing the quality of protein-docking algorithms seems straightforward. In principle, it would suffice to apply the docking algorithm to a number of test cases for which the complex structure has been determined experimentally and compare the generated conformations with the known

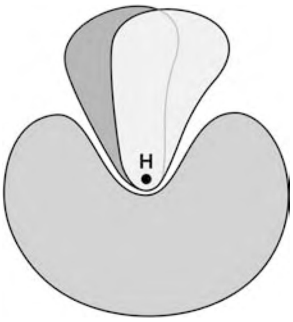


Figure 8 RMSD is not a well-suited measure for the similarity of protein–protein complexes. Small rotations of one of the proteins about a point close to the interface can lead to large changes in RMSD, although most of the interface contacts are conserved.

complex structures. Unfortunately, two problems arise. First, the choice of the test complexes has an enormous influence on the quality of the result. Some protein complexes are inherently "simple" and others inherently "hard", typically due to the degree of flexibility. Second, there is no canonical measure for the deviation of the generated structures from the native ones [110]. The most popular method is the computation of the RMSD of a certain set of atoms, e.g. the C_α atoms of the backbone. The usefulness of the RMSD for docking evaluation has been questioned in the literature [119]. One of the problems associated with RMSD is illustrated in Figure 8. While a small rotation of one of the proteins about a point close to the interface preserves most of the interface contacts, it results in a large change in RMSD. Alternative quality measures thus often focus in some way on the interface region. The interfacial RMSD, for example, computes the RMSD only for binding site residues. A conceptually different measure is the fraction of native contacts, which tends to be more stable for partially correct predictions [119]. It counts the number of native contacts present in the candidate structure.

Assessing the quality of docking predictions is further complicated by the relatively low number of protein complexes with known bound and unbound structures. Since a considerable number of docking examples is typically needed to tune the docking algorithm and fit its parameters, the real *test set* is usually quite small. It is thus often unclear how well the predictions of a given algorithm generalize to completely unknown complexes. Apart from the above-mentioned more technical aspects, comparing the results of different docking algorithms and assessing their quality is severely complicated by a further, more subtle difficulty: typically, during the development and testing of protein-docking algorithms, no clear distinction is made between test and training sets. This means that in many cases, the structures that are used to evaluate the quality of a given docking algorithm were also employed during its development. Hence, it cannot be ruled out that the algorithm has been overtrained, i.e. that it shows good performance on the training set only.

To overcome these difficulties, the protein-docking community has established a regular blind prediction challenge called CAPRI which is similar in spirit to the CASP competition in protein structure prediction. CAPRI is organized in so-called rounds. For each round, crystallographers provide possible docking targets, the structures of which they have already determined, but not published. The organizers choose an interesting subset of these targets and forward the structures of the individual proteins – either in their bound or unbound (for some cases, only a homologous unbound structure is used as input, further complicating the prediction) conformation, depending on the target – to the participating groups. Each group may then submit a certain number of predictions for each target. Finally, the predictions are evaluated by the organizers and the results are published.

Detailed evaluations of the different CAPRI rounds can be found in the corresponding papers and on the CAPRI website [35]. Here, we will exemplarily discuss the evaluation of round three as taken from [116]. CAPRI round three presented only two different targets: an epidermal growth factor-like domain of laminin and nidogen-G3 and a wild-type homodimer of the phosphoenolpyruvate:sugar phosphotransferase system (PTS) regulation domain from LicT [55]. Of these two targets, the first one was considered the easier problem: nidogen was used in its bound form (but randomly oriented) and laminin, even though used in the unbound form, was very similar in structure to the bound conformation. Out of the 179 submitted predictions for this target, 27 were of acceptable or better quality. Only two models out of these 27 were considered high quality, i.e. they had more than 50% of the native contacts correctly predicted. The second target was considerably more difficult. Here, the challenge consisted in predicting the wild-type homodimer from the structure of the subunit of a double mutant, which produces an activated form of the homodimer with widely different backbone (RMSD of around 12 Å). This large degree of flexibility posed severe difficulties to the docking algorithms and, indeed, the prediction score in general was quite poor [116], with only one acceptable prediction.

These results already indicate that for certain complex structures, protein–protein docking is still a major challenge. In fact, the level of difficulty depends mostly on the type of structure considered. Vajda and Camacho [110] analyzed strengths and weaknesses of current docking techniques on a benchmark set of 52 different complexes [19] (this benchmark set has since been extended in Ref. [87]). Interestingly, it turned out that the complexes in the test set can be assigned a degree of difficulty that is independent of the docking algorithm applied. The whole test set decomposes into five categories of increasing difficulty based on three structural properties: the degree of flexibility, the area of the complex interface (ΔSASA, the change in solvent accessible surface area during complex separation) and the magnitude of the desolvation effect upon complex formation. The simplest class of docking examples, the so-called type I complexes, is composed of protein–protein complexes with low degree of flexibility, a relatively large interface area of $1400 < \Delta\text{SASA} < 2000$ Å^2 and a strong desolvation effect below -4 kcal mol^{-1}. This class consists almost completely of enzyme-inhibitor complexes. If the hydrophobic effect is weaker than for the complexes of type I, protein–protein docking can still be expected to work successfully as long as the interface area is large, $\Delta\text{SASA} > 2000$ Å^2, and the conformational changes during docking are only moderate. These complexes form class II, which consists mainly of protein pairs with strong functional coupling and of enzyme–inhibitor complexes.

The docking problem becomes more difficult if neither the hydrophobic effect nor the interface area are particularly large. In this case, which is called type III, the docking results often depend sensitively on the coordinates of the input structures. An important type of complexes that often falls into this class is antibody–antigen pairs. If in addition the interface area is particularly small, $\Delta\text{SASA} < 1400$ Å^2, docking can no longer be expected to work successfully. Indeed, the results of all docking algorithms for this type IV complexes are generally poor. Interestingly, Vajda and Camacho noticed that many docking algorithms still produce several true positives in the structure generation stage for type IV complexes. However, these are usually lost in the filtering or scoring steps of the algorithm.

The hardest problem for all current docking algorithms, though, is those complexes which undergo large conformational changes during docking. This effect often occurs if the interface area is larger than $\Delta\text{SASA} > 2000$ Å^2: large contact areas give rise to a large number of interactions and these interactions are able to induce massive conformational changes. Thus, it is sometimes hard to distinguish between the relatively easy type II complexes and the extremely hard type V complexes. Transient protein complexes involved in signal transduction typically fall into this class.

Having defined these five classes of increasing difficulty, Vajda and Camacho evaluated whether the results of the CAPRI competition fit into this scheme. Indeed, it turned out that the decomposition of the CAPRI targets into classes I–V explained the performance of the different algorithms very well. In general, all algorithms predict the correct structure for type I and II complexes. The results for type III complexes varied, but were consistently better for those with negative desolvation energy, hence for complexes closer to class I. Type IV complexes lead to poor docking results, while docking seems to fail for type V. This study shows an important difference in the source of error for the different types of complexes: while for class III and

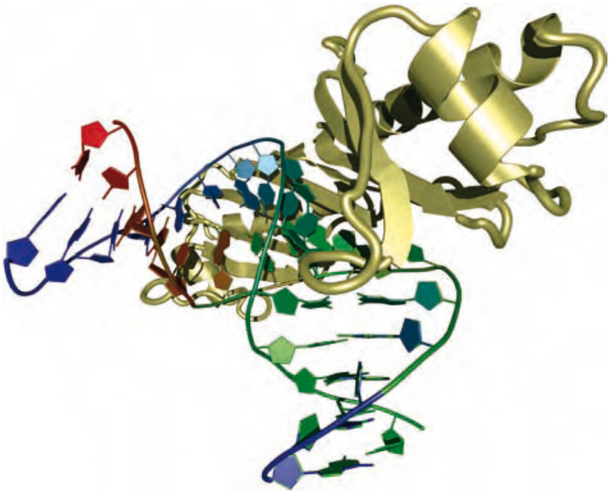


Figure 9 As can be seen from this crystal structure of the TATA box-binding protein bound to double-stranded DNA (PDB ID 1CDW [93]), proteins can induce massive conformational changes of the DNA. In this case, the DNA is nearly bent to a right angle.

IV, we can expect major improvements by changing scoring functions and SCP algorithms, type V complexes will need new techniques for handling flexibility. Since in these cases a rigid structure generation phase will typically not produce approximations to the native structure, improved re-ranking methods in the later stages will come too late. The only possible remedy to this shortcoming would be the flexibilization of the structure generation phase.

3 Protein–DNA Interactions

3.1 Peculiarities of Protein–DNA Binding

Interactions of proteins with nucleic acids differ in some fundamental aspects from protein–protein interactions. The most prominent structural peculiarity is the negatively charged phosphodiester backbone of nucleic acids. Proteins binding to DNA have to compensate this charge, which typically results in a large number of positively charged lysine and arginine residues in the protein–DNA interface. Hence, electrostatics, solvent effects and counterions play a much more important role in modeling protein–DNA interactions than they do in protein–protein interactions.

The second key difference lies in the flexibility of nucleic acids. DNA and, in particular, RNA have quite flexible backbones, which often undergo bending or unwinding upon binding (Figure 9). Apart from changes in the backbone conformation, individual bases or base pairs can flip out of the double helix and interact directly with the protein. To date, no algorithm is truly able to handle this degree of flexibility. However, there are some proteins, e.g. zinc finger proteins, that bind to DNA without inducing significant changes in DNA structure. In these cases, complex structures can be successfully predicted with approaches quite similar to those applicable to protein–protein docking. However, scoring functions used in protein–protein docking have to be adapted to the different types of interactions present in the protein–DNA case. The construction of knowledge-based potentials for protein–DNA interactions is further complicated by the comparatively small number of crystal structures available; only a minor fraction of the complex structures found in the Protein Data Bank (PDB) contains nucleic acids (about 900 structures as of early 2006).

While the electrostatic interaction of the protein with the DNA backbone is the major contributing factor towards protein–DNA complex stability, specificity is mostly conveyed through hydrogen bonds between protein side-chains and bases. This makes the prediction of the exact binding position, i.e. the sequence motif a specific protein binds to, much harder than the prediction of the binding site of the protein. This difficulty is based on two facts. (i) The relative differences in binding free energy between two DNA sequences are often quite low with differences in the range of a few $\mathrm{kcal\,mol^{-1}}$. Hence, very accurate scoring functions are required. (ii) The hydrogen bond patterns conveying specificity often include water-mediated hydrogen bonds, which are difficult to predict and to model.

The ability to accurately predict relative binding affinities for different sequences (i.e. the specificity) is an important goal in protein–DNA docking. This allows the development of rapid sequence-based prediction methods based on structural data and is also an essential tool for the design of tailor-made transcription factors [50].

3.2 Algorithmic Techniques

As protein–DNA docking is currently restricted to complexes with little conformational flexibility, it basically employs the same techniques used in rigid-body protein–protein docking. We will now describe some of these docking techniques together with the scoring functions.

3.2.1 Correlation Techniques

Several approaches have been proposed that use correlation techniques for protein–DNA docking [3, 99]. These approaches use established tools from protein–protein docking to perform a RBD, and then filter and refine the results. We will discuss the work by Aloy and coworkers [3] in more detail.

Their docking algorithm applies a standard correlation-based RBD method (FTDOCK) used with slightly different parameters and electrostatics. As has been mentioned above, the binding specificity for many protein–DNA complexes is only marginally influenced by the DNA backbone charges. Aloy and coworkers account for this by dampening the backbone charges. The correlation docking results in a set of initial candidate structures, which are then filtered with respect to geometric constraints. These constraints are derived from experimentally determined DNA–protein contacts. The remaining candidates are finally re-scored using a knowledge-based potential. Although the docking does not account for flexibility, it turns out to be quite successful even when used with unbound protein structures. Of a test set of eight complexes, two complexes were indeed predicted correctly (below 4 Å RMSD for the structure on rank one) and for five out of the eight a good approximation was at least among the top ten structures.

3.2.2 Monte Carlo Techniques

Knegtel and coworkers [60] proposed a Monte Carlo algorithm for protein–DNA docking. MONTY uses a simple scoring function considering hydrogen bonds, salt bridges, van der Waals contacts and steric collisions. This scoring function is used in a Monte Carlo framework to identify minimum energy conformations. MONTY considers the six degrees of freedom arising from protein rotation and translation, and additional degrees of freedom representing flexible side-chains. Repeated runs of the Monte Carlo simulation yield different candidate structures. The approach was tested on the complex of 434 Cro and double-stranded DNA. For three different X-ray structures, MONTY was able to identify a good approximation of the true complex structure (below 4 Å RMSD). However, the algorithm started from the bound structures and did not account for DNA flexibility.

In a second study, Knegtel and coworkers [61] extended MONTY to include DNA backbone flexibility as well. They describe the DNA backbone through a parametric equation and allow for unwinding or tightening of the helix during the simulation. Helix flexibility is achieved by changing the parameters of the parametric equation in small steps. Base pairs are then relaxed to adjust to the new backbone conformation. It turns out that adding DNA backbone flexibility indeed increases the number of correctly retrieved hydrogen bonds. It was also shown that inclusion of experimental data improves the predictions. To this end, Knegtel and coworkers included harmonic potentials for NOE pairs or energy bonuses for correctly retrieved experimentally confirmed contacts. These contacts can be obtained from arbitrary experiments, e.g. from mutagenesis.

A hybrid geometric hashing, minimization, and Monte Carlo approach was proposed by Deng and coworkers [17, 26]. Their algorithm is based on the assumption that the hydrogen bonding is the key factor in protein–DNA interaction. Hence, they designed an interaction potential accounting for hydrogen bonding and van der Waals interactions only. The piecewise defined function resembles a Lennard–Jones potential, but also includes angular contributions for hydrogen bonds. The different phases of the algorithm use different levels of approximation for this potential. In the initial geometric hashing phase of the algorithm, the potential is approximated as a square-well potential, which in fact just counts hydrogen bonds between DNA and protein. Matching quadruples of interaction points between protein and DNA creates a list of transformations ensuring at least four hydrogen bond matches between protein and DNA. The second phase of the algorithm refines the square-well approximation to a harmonic potential. For this quadratic approximation one can easily derive a closed-form solution for the optimal transformation minimizing the interaction energy. The optimal solution for the quadratic scoring function is then used as the starting point for a Monte Carlo minimization using the full Lennard–Jones potential in the third phase. For a number of well-known DNA-binding proteins they could predict the correct (i.e. highest binding) sequence motif.

3.3 Scoring Functions

Mandel-Gutfreund and coworkers [79, 80] proposed a simple residue-based contact potential based on a set of 53 protein–DNA structures. They applied this potential to sequence variants of a DNA–zinc finger complex. To identify binding motifs, they scanned the upstream regions of a number of genes. The sequence of the DNA was changed to all potential motifs occuring in these upstream regions and these motifs were then scored with the contact potential. Although the potential is quite simple, it allows structure-based predictions of the binding sites with reasonable chance of success.

Zhang and coworkers [120] proposed a knowledge-based energy function for protein–DNA and protein–ligand complexes. The potential differs from other knowledge-based approaches through the choice of the reference state. The distance-scaled, finite, ideal-gas references state (DFIRE [122, 123]) avoids some the problems related to other commonly used reference states. As in similar knowledge-based potentials, the DFIRE potential is derived from binned contact frequencies in a database of 200 protein–ligand complexes. The potential was validated on 45 protein–DNA complexes from the PDB with known binding free energies. On this data set the scoring function yielded an excellent agreement of predicted and experimental binding free energies, although no protein–protein and protein–DNA complexes were used to derive the potential. The potential is thus very robust and generalizes well to other types of structures.

Several authors have suggested that the directionality of the hydrogen bonds is an essential feature, which is not properly captured by these simple potentials. They thus extend the idea of knowledge-based potentials to include angular contributions besides the radial contributions as well. Kono and Sarai [64] proposed a potential based on the spatial distribution of amino acid atoms around the base pair. They analyzed 52 protein–DNA complex structures and counted the frequency of occurence of hydrogen bond donors and acceptors in defined spatial positions (relative to the fixed base) by summing up observations on a 3-D cubic grid. This type of potential contains an angular contribution through the relative position of a grid point with respect to the base. The potential was shown to successfully predict the DNA target sequences for the MATα1/α2 transcription factor.

Another knowledge-based potential was proposed by Ge and coworkers [42]. It is specific for the interaction with base pairs in the DNA major and minor grooves. Following a similar idea as Kono & Sarai, they superimpose DNA base pairs from high-quality crystallographic data. From this data, they compile a set of observed positions for water molecules and amino acid atoms relative to the base pairs. This set of points is then approximated by 3-D ellipsoids, which represent areas of favorable interaction with the base pair. Interaction energies with the base pairs are estimated as simple harmonic potentials between protein atoms and DNA ellipsoids. This energy function is complemented by a simple steric clash test (overlapping conformations are excluded) and performed quite well for a number of ligand–DNA complexes.

4 Conclusion

Protein–protein docking has seen many improvements over the last decade, which are highlighted by the results of blind predictions in the CAPRI competition. At the same time, the blind prediction experiments proved that protein–protein docking is still a largely unsolved problem. A careful review of the failings and problems of the numerous techniques and applications described above reveals some fundamental problems that need to be addressed in the future.

The first key problem still is *protein flexibility*. There are now several techniques available for handling flexible side-chains and domain movements in protein docking. Despite the progress made, none of these approaches addresses protein flexibility in its entirety. Except for a few stochastic approaches, e.g. Monte Carlo approaches, we have not yet managed to fuse the two first steps of the docking – structure generation and realizing flexibility – into a single combined step.

Scoring functions are the second key problem in protein–protein docking. In particular, solvent effects and entropic contributions remain challenging problems. We have seen many improvements in this area, but successful scoring functions are still computationally quite demanding. Over recent years, scoring functions have continually improved. Nevertheless, we are far from the point where one single scoring function can reliably distinguish the native complex structure from decoys.

If protein–protein docking is a difficult problem, then DNA–protein docking is much more difficult by far. While the fundamental problem is nearly identical, the development in this area is still a few years behind in every aspect. Fewer crystal structures are available, therefore our understanding of the protein–DNA interaction is not yet as far advanced as our understanding of the protein–protein interaction. The two key problems of protein–protein docking, i.e. flexibility and accurate scoring, occur in protein–DNA docking as well. In fact, both problems are more severe in this case. The huge conformational changes induced to DNA upon protein binding are still beyond what any docking algorithm is able to handle. The same is true for the accurate scoring, where the peculiar effects of DNA backbone charges and the effect of counterions are still major problems.

Protein–protein and protein–DNA docking will clearly remain exciting and important topics in the future. The rising interest in a structure-based understanding of protein interaction networks has given the field new impulses. Structure-based systems biology tries to understand large, complex systems based on molecular interactions [4, 100]. While the new data obtained from structural genomics initiatives has taught us a lot about interactions, this way of thinking opened up new application areas for protein–protein docking. For example, protein–protein docking can help to validate or predict protein interaction networks. Protein–DNA docking might play a similar role for GRNs, however, we are still a long way from being able to predict these networks based on structure alone.

The key challenges of the next years lie in the development of improved scoring functions and full flexibility for both proteins and DNA. Better scoring functions would not only yield an accurate ranking of the candidate structures, but also a good prediction of the absolute value of the binding free energy. This is a challenging task indeed. It remains to be seen whether it can be reached through any of the current methods. All thermodynamic quantities, including the binding free energy, are defined for an ensemble of structures. It would thus seem obvious that simulation-based methods are more approriate for predicting these quantities than the current methods based on a single structure only. However, this implies massive investments in terms of compute power. An interesting alternative to more accurate scoring is “data-driven docking”. Here, we can expect numerous new techniques to be integrated into docking over the next years. Basically any technique producing structure-based information of some kind can be integrated into docking approaches. Hopefully, these different efforts will converge into one consistent framework for data-driven docking in the future. Future algorithms will also have to account for side-chain and backbone flexibility simultaneously. For the case of protein–DNA docking we will probably need entirely different approaches to account for full flexiblity. These challenges will clearly keep the docking community busy for the next years and keep the CAPRI competitions exciting.

Acknowledgments

The authors wish to thank Andreas Moll, Anne Dehof and Bettina Leonhardt for their assistance with preparing the figures, and Pierre Dönnes, Annette Höglund and Dr. Dirk Neumann for critical reading of the manuscript. All images of molecular structures were generated using BALLView [88].

References

1 R. Abagyan and M. Totrov. Biased probability Monte Carlo conformational searches and electrostatic calculations for peptides and proteins. *J. Mol. Biol.* 235:983–1002, 1994.

2 F. Ackermann, G. Herrmann, S. Posch and G. Sagerer. 3D-Segmentierungstechniken und vektorwertige Bewertungsfunktionen für symbolisches Protein–Protein-Docking. In D. Schomburg and U. Lessel (eds.), *Bioinformatics: From Nucleic Acids and Proteins to Cell Metabolism.* GBF, Braunschweig, 105–24, 1995.

3 P. Aloy, G. Moont, H. A. Gabb, E. Querol, F. X. Aviles, and M. J. E. Sternberg. Modelling repressor proteins docking to DNA. *Proteins* 33:535–49, 1998.

4 P. Aloy and R. B. Russell. Structure-based systems biology: a zoom lens for the cell. *FEBS Lett.* 579:1854–8, 2005.

5 E. Althaus, O. Kohlbacher, H.-P. Lenhof and P. Müller. A combinatorial approach to protein docking with flexible side-chains. *Proc. RECOMB* 3:15–24, 2000.

6 E. Althaus, O. Kohlbacher, H.-P. Lenhof and P. Müller. A combinatorial approach to protein docking with flexible side-chains. *J. Comput. Biol.* 9:597–612, 2002.

7 G. S. Anand, D. Law, J. G. Mandell, A. N. Snead, I. Tsigelny, S. S. Taylor, L. F. Ten Eyck and E. A. Komives. Identification of the protein kinase A regulatory RIα-catalytic subunit interface by amide H/2H exchange and protein docking. *Proc. Natl Acad. Sci. USA* 100:13264–9, 2003.

8 M. R. Arkin, M. Randal and W. L. DeLano, et al. Binding of small molecules to an adaptive protein–protein interface. *Proc. Natl Acad. Sci. USA* 100:1603–8, 2003.

9 G. Ausiello, G. Cesareni and M. Helmer-Citterich. ESCHER: a new docking procedure applied to the reconstruction of protein tertiary structure. *Proteins* 28:556–67, 1997.

10 D. J. Bacon and J. Moult. Docking by least-squares fitting of molecular surface patterns. *J. Mol. Biol.* 225:849–58, 1992.

11 E. Ben-Zeev and M. Eisenstein. Weighted geometric docking:

incorporating external information in the rotation-translation scan. *Proteins* 52:24–7, 2003.

12 E. BEN-ZEEV, R. ZARIVACH, M. SHOHAM, A. YONATH AND M. EISENSTEIN. Prediction of the structure of the complex between the 30S ribosomal subunit and colicin E3 via weighted-geometric docking. *J. Biomol. Struct. Dyn.* 20:669–76, 2003.

13 M. J. BETTS AND M. J. STERNBERG. An analysis of conformational changes on protein–protein association: implications for predictive docking. *Protein Eng.* 12:271–83, 1999.

14 J. BRICKMANN. Linguistic variables in the molecular recognition problem. In D. H. Rouvray (ed.), *Fuzzy Logic Chemistry*. Academic, San Diego, CA, 225–247, 1997.

15 N. BROOIJMANS AND I. D. KUNTZ. Molecular recognition and docking algorithms. *Annu. Rev. Biophys. Biomol. Struct.* 32:335–73, 2003.

16 R. E. BRUCCOLERI AND J. NOVOTNY. Antibody modeling using the conformational search program CONGEN. *Immunomethods* 1:96–106, 1992.

17 G. CAMPBELL, Y. DENG, J. GLIMM, Y. WANG, Q. YU, M. EISENBERG AND A. GROLLMAN. Analysis and prediction of hydrogen bonding in protein-DNA complexes using parallel processors. *J. Comput. Chem.* 17:1714–1725, 1996.

18 B. CHAZELLE, C. KINGSFORD AND M. SINGH. A semidefinite-programming approach to side-chain positioning with new rounding strategies. *INFORMS J. Comput.* 16:380–92, 2004.

19 R. CHEN, J. MINTSERIS, J. JANIN AND Z. WENG. A protein–protein docking benchmark. *Proteins* 52:88–91, 2003.

20 J. CHERFILS, S. DUQUERROY AND J. JANIN. Protein–protein recognition analyzed by docking simulation. *Proteins* 11:271–80, 1991.

21 M. L. CONNOLLY. Solvent-accessible surfaces of proteins and nucleic acids. *Science*, 221:709–13, 1983.

22 M. L. CONNOLLY. Shape complementarity at the hemoglobin α1β1 subunit interface. *Biopolymers* 25:1229–47, 1986.

23 M. L. CONNOLLY. Analytical molecular surface calculation. *J. Appl. Crystallogr.* 16:548–58, 1983.

24 T. P. CREAMER AND G. D. ROSE. Side-chain entropy opposes α-helix formation but rationalizes experimentally determined helix-forming propensities. *Proc. Natl Acad. Sci. USA* 89:5937–41, 1992.

25 T. P. CREAMER AND G. D. ROSE. α-helix-forming propensities in peptides and proteins. *Proteins* 19:85–97, 1994.

26 Y. DENG, J. GLIMM, Y. WANG, A. KOROBKA, M. EISENBERG AND A. P. GROLLMAN. Prediction of protein binding to DNA in the presence of water-mediated hydrogen bonds. *J. Mol. Model.* 5:125–133, 1999.

27 J. DESMET, M. DE MAEYER, B. HAZES AND I. LASTERS. The dead end elimination theorem and its use in protein side-chain positioning. *Nature* 356:539–542, 1992.

28 J. DESMET, M. DE MAEYER AND I. LASTERS. The dead-end elimination theorem: a new approach to the side-chain packing problem. In *The Protein Folding Problem and Tertiary Structure Prediction.* Birkhäuser, Basel, 307–337, 1994.

29 J. DESMET, M. DE MAEYER AND I. LASTERS. 1997. Theoretical and algorithmical optimization of the dead-end elimination theorem. *Pac. Symp. Biocomput.*: 122–33.

30 C. DOMINGUEZ, R. BOELENS AND A. M. J. J. BONVIN. HADDOCK: a protein–protein docking approach based on biochemical or biophysical information. *J. Am. Chem. Soc.* 125:1731–7, 2003.

31 R. L. DUNBRACK AND F. E. COHEN. Bayesian statistical analysis of protein side-chain rotamer preferences. *Protein Sci.* 6:1661–81, 1997.

32 P. EHRLICH. Chemotherapeutics: scientific principles, methods, and results. *Lancet* 182:445–51, 1913.

33 M. EISENSTEIN AND E. KATCHALSKI-KATZIR. On proteins, grids, correlations, and docking. *C. R. Biol.* 327:409–20, 2004.

34 O. ERIKSSON, Y. ZHOU AND A. ELOFSSON. Side chain-positioning as an integer programming problem. In Proc. 1st Workshop on Algorithms in Bioinformatics, Aarhus, Denmark, 2149:129–41, 2001.

35 EMBL/EBI European Bioinformatics Institute. CAPRI community-wide experiment on the comparative evaluation of protein–protein docking for structure prediction. http://capri.ebi.ac.uk/.

36 T. E. Exner and J. Brickmann. New docking algorithm based on fuzzy set theory. *J. Mol. Model.* 3:321–24, 1997.

37 F. A. Momany, R. F. McGuire, A. W. Burgess and H. A. Scheraga. Energy parameters in polypeptides. VII. Geometric parameters, partial atomic charges, nonbonded interactions, hydrogen bond interactions, and intrinsic torsional potentials for the naturally occurring amino acids. *J. Phys. Chem.* 79:2361–81, 1975.

38 D. Fischer, S. L. Lin, H. L. Wolfson and R. Nussinov. A geometry-based suite of molecular docking processes. *J. Mol. Biol.* 248:459–77, 1995.

39 E. Fischer. Einfluss der Configuration auf die Wirkung der Enzyme. *Ber. Dt. Chem. Ges.* 27:2985–2993, 1894.

40 H. A. Gabb, R. M. Jackson and M. J. E. Sternberg. Modelling protein docking using shape complementarity, electrostatics and biochemical information. *J. Mol. Biol.* 272:106–120, 1997.

41 E. J. Gardiner, P. Willett and P. J. Artymiuk. Protein docking using a genetic algorithm. *Proteins* 44:44–56, 2001.

42 W. Ge, B. Schneider and W. K. Olson. Knowledge-based elastic potentials for docking drugs or proteins with nucleic acids. *Biophys J.* 88:1166–90, 2005.

43 M. K. Gilson and B. Honig. Calculation of the total electrostatic energy of a macromolecular system: solvation energies, binding energies, and conformational analysis. *Proteins* 4:7–18, 1988.

44 R. F. Goldstein. Efficient rotamer elimination applied to protein side-chains and related spin glasses. *Biophys. J.* 66:1335–40, 1994.

45 D. B. Gordon, G. K. Hom, S. L. Mayo and N. A. Pierce. Exact rotamer optimization for protein design. *J. Comput. Chem.* 24:232–43, 2003.

46 D. B. Gordon and S. L. Mayo. Radical performance enhancements for combinatorial optimization algorithms based on the dead-end elimination theorem. *J. Comput. Chem.* 19:1505–1514, 1998.

47 P. E. Hart, N. J. Nilsson and B. A. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. on SSC*, 4:100–114, 1968.

48 A. Heifetz, E. Katchalski-Katzir and M. Eisenstein. Electrostatics in protein–protein docking. *Protein Sci.* 11:571–87, 2002.

49 M. Helmer-Citterich and A. Tramontano. PUZZLE: a new method for automated protein docking based on surface shape complementarity. *J. Mol. Biol.* 235:1021–31, 1994.

50 A. Höglund and O. Kohlbacher. From sequence to structure and back again: approaches for predicting protein-DNA binding. *Proteome Sci.* 2:3, 2004.

51 L. Holm and C. Sander. Database algorithm for generating protein backbone and side-chain co-ordinates from a C_α trace application to model building and detection of co-ordinate errors. *J. Mol. Biol.* 218:183–94, 1991.

52 L. Holm and C. Sander. Fast and simple Monte Carlo algorithm for side-chain optimization in proteins: application to model building by homology. *Proteins* 14:213–23, 1992.

53 J. K. Hwang and W. F. Liao. Side-chain prediction by neural networks and simulated annealing optimization. *Protein Eng.* 8:363–70, 1995.

54 R. M. Jackson and M. J. Sternberg. A continuum model for protein–protein interactions: application to the docking problem. *J. Mol. Biol.* 250:258–75, 1995.

55 J. Janin. The targets of CAPRI rounds 3-5. *Proteins* 60:170–5, 2005.

56 F. Jiang and S. H. Kim. 'Soft docking' matching of molecular surface cubes. *J. Mol. Biol.* 219:79–102, 1991.

57 N. Kasinos, G. A. Lilley, N. Subbarao and I. Haneef. A robust and efficient

automated docking algorithm for molecular recognition. *Protein Eng.* 5:69–75, 1992.

58 E. Katchalski-Katzir, I. Shariv, M. Eisenstein, A. A. Friesem, C. Aflalo and I. A. Vakser. Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proc. Natl Acad. Sci. USA* 89:2195–9, 1992.

59 C. L. Kingsford, B. Chazelle and M. Singh. Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics* 21:1028–36, 2005.

60 R. M. A. Knegtel, J Antoon, C. Rullmann, R. Boelens and R. Kaptein. MONTY: a Monte Carlo approach to protein-DNA recognition. *J. Mol. Biol.* 235:318–24, 1994.

61 R. M. A. Knegtel, R. Boelens and R. Kaptein. Monte Carlo docking of protein-DNA complexes: incorporation of DNA flexibility and experimental data. *Protein Eng.* 7:761–67, 1994.

62 P. Koehl and M. Delarue. Application of a self-consistent mean field theory to predict protein side-chains conformation and estimate their conformational entropy. *J. Mol. Biol.* 239:249–275, 1994.

63 O. Kohlbacher, A. Burchardt, A. Moll, A. Hildebrandt, P. Bayer and H. P. Lenhof. Structure prediction of protein complexes by an NMR-based protein docking algorithm. *J. Biomol. NMR* 20:15–21, 2001.

64 H. Kono and A. Sarai. Structure-based prediction of DNA target sites by regulatory proteins. *Proteins* 35:114–31, 1999.

65 D. E. Jr. Koshland. Application of a theory of enzyme specificity to protein synthesis. *Proc. Natl Acad. Sci. USA*, 44:98–104, 1958.

66 I. D. Kuntz, J. M. Blaney, S. J. Oatley, R. Langridge and T. E. Ferrin. A geometric approach to macromolecule-ligand interactions. *J. Mol. Biol.* 161:269–88, 1982.

67 I. Lasters, M. De Maeyer and J. Desmet. Enhanced dead-end elimination in the search for the global minimum energy conformation of a collection of protein side-chains. *Protein Eng.* 8:815–22, 1995.

68 C. A. Laughton. Prediction of protein side-chain conformations from local three-dimensional homology relationships. *J. Mol. Biol.* 235:1088–97, 1994.

69 A. R. Leach. Ligand docking to proteins with discrete side-chain flexibility. *J. Mol. Biol.* 235:345–56, 1994.

70 A. R. Leach and A. P. Lemon. Exploring the conformational space of protein side-chains using dead-end elimination and the A^* algorithm. *Proteins* 33:227–39, 1998.

71 C. Lee. Predicting protein mutant energetics by self-consistent ensemble optimization. *J. Mol. Biol.* 236:918–39, 1994.

72 C. Lee and S. Subbiah. Prediction of protein side-chain conformation by packing optimization. *J. Mol. Biol.* 217:373–88, 1991.

73 H.-P. Lenhof. New contact measures for the protein docking problem. *Proc. RECOMB* 1:182–91, 1997.

74 D. Levine, M. Facello, P. Hallstrom, G. Reeder, B. Walenz and F. Stevens. STALK: an interactive system for virtual molecular docking. *IEEE Trans. Comp. Sci. Eng.* 4: 55–65, 1997.

75 M. Levitt. Accurate modeling of protein conformation by automatic segment matching. *J. Mol. Biol.* 226:507–33, 1992.

76 S. L. Lin, R. Nussinov, D. Fischer and H. J. Wolfson. Molecular surface representations by sparse critical points. *Proteins* 18:94–101, 1994.

77 S. Liu, C. Zhang, H. Zhou and Y. Zhou. A physical reference state unifies the structure-derived potential of mean force for protein folding and binding. *Proteins* 56:93–101, 2004.

78 L. L. Looger and H. W. Hellinga. Generalized dead-end elimination algorithms make large-scale protein side-chain structure prediction tractable: implications for protein design and structural genomics. *J. Mol. Biol.* 307:429–45, 2001.

79 Y. Mandel-Gutfreund, A. Baron and H. Margalit. A structure-based approach for prediction of protein binding sites in gene upstream regions. *Pac. Symp. Biocomput.*: 139–50, 2001.

80 Y. Mandel-Gutfreund and H. Margalit. Quantitative parameters for amino acid–base interaction: implications for prediction of protein–DNA binding sites. *Nucleic Acids Res.* 26:2306–12, 1998.

81 J. G. Mandell, V. A. Roberts, M. E. Pique, V. Kotlovyi, J. C. Mitchell, E. Nelson, I. Tsigelny and L. F. Ten Eyck. Protein docking using continuum electrostatics and geometric fit. *Protein Eng.* 14:105–13, 2001.

82 A. J. McCoy, V. Chandana Epa and P. M. Colman. Electrostatic complementarity at protein/protein interfaces. *J. Mol. Biol.* 268:570–84, 1997.

83 A. D. McLachlan. Gene duplications in the structural evolution of chymotrypsin. *J. Mol. Biol.* 128:49–79, 1979.

84 J. Mendes, A. M. Baptista, M. A. Carrondo and C. M. Soares. Improved modeling of side-chains in proteins with rotamer-based methods: a flexible rotamer model. *Proteins* 37:530–43, 1999.

85 J. Mendes, C. M. Soares and M. A. Carrondo. Improvement of side-chain modeling in proteins with the self-consistent mean field theory method based on an analysis of the factors influencing prediction. *Biopolymers* 50:111–31, 1999.

86 N. Metropolis and S. Ulam. The Monte Carlo Method. *J. Am. Stat. Assoc.* 44:335, 1949.

87 J. Mintseris, K. Wiehe, B. Pierce, R. Anderson, R. Chen, J. Janin and Z. Weng. Protein–protein docking benchmark 2.0: an update. *Proteins* 60:214–6, 2005.

88 A. Moll, A. Hildebrandt, H.-P. Lenhof and O. Kohlbacher. BALLView: a tool for research and education in molecular modeling. *Bioinformatics* 22:365–6, 2006.

89 J. B. Moon and W. J. Howe. Computer design of bioactive molecules: a method for receptor-based de novo ligand design. *Proteins* 11:314–28, 1991.

90 G. Moont, H. A. Gabb and M. J. Sternberg. Use of pair potentials across protein interfaces in screening predicted docked complexes. *Proteins* 35:364–73, 1999.

91 J. Murphy, D. W. Gatchell, J. C. Prasad and S. Vajda. Combination of scoring functions improves discrimination in protein–protein docking. *Proteins* 53:840–54, 2003.

92 G. Némethy, M. S. Pottle and H. A. Scheraga. Energy parameters in polypeptides. 9. Updating of geometrical parameters, nonbonded interactions and hydrogen bond interactions for the naturally occurring amino acids. *J. Phys. Chem.* 87:1883–7, 1983.

93 D. B. Nikolov, H. Chen, E. D. Halay, A. Hoffman, R. G. Roeder and S.K. Burley. Crystal structure of a human TATA box-binding protein/TATA element complex. *Proc. Natl Acad. Sci. USA* 93:4862–7, 1996.

94 R. Norel, S. L. Lin, D. Xu, H. J. Wolfson and R. Nussinov. Molecular surface variability and induced conformational changes upon protein–protein association. In R. H. Sarma and M. H. Sarma (eds.), *Structure, Motion, Interaction and Expression of Biological Macromolecules. Proceedings of the 10th Conversation.* Schenectady, NY, 33–51, 1998.

95 P. N. Palma, L. Krippahl, J. E. Wampler and J. J. Moura. BiGGER: a new (soft) docking algorithm for predicting protein interactions. *Proteins* 39:372–84, 2000.

96 N. A. Pierce and E. Winfree. Protein design is NP-hard. *Protein Eng.* 15:779–82, 2002.

97 J. W. Ponder and F. M. Richards. Tertiary templates for proteins – use of packing criteria in the enumeration of allowed sequences for different structural classes. *J. Mol.Biol.* 193:775–791, 1987.

98 M. Rarey, B. Kramer, T. Lengauer and G. Klebe. A fast flexible docking method using an incremental construction algorithm. *J. Mol. Biol.* 261:470–89, 1996.

99 V. A. Roberts, D. A. Case and V. Tsui. Predicting interactions of winged-helix transcription factors with DNA. *Proteins* 57:172–87, 2004.

100 R. B. Russell, F. Alber, P. Aloy, F. P. Davis, D. Korkin, M. Pichaud, M. Topf and A. Sali. A structural perspective on protein–protein interactions. *Curr. Opin. Struct. Biol.* 14:313–24, 2004.

101 B. Sandak, R. Nussinov and H. J. Wolfson. An automated computer vision and robotics-based technique for 3-D flexible biomolecular docking and matching. *Comput. Appl. Biosci.* 11:87–99, 1995.

102 B. Sandak, R. Nussinov and H. J. Wolfson. A method for biomolecular structural recognition and docking allowing conformational flexibility. *J. Comput. Biol.* 5:631–54, 1998.

103 D. Schneidman-Duhovny, R. Nussinov and H. J. Wolfson. Predicting molecular interactions in silico: II. protein–protein and protein–drug docking. *Curr. Med. Chem.* 11:91–107, 2004.

104 B. A. Shoemaker, J. J. Portman and P. G. Wolynes. Speeding molecular recognition by using the folding funnel: the fly-casting mechanism. *Proc. Natl Acad. Sci. USA* 97:8868–73, 2000.

105 B. K. Shoichet, D. L. Bodian and I. D. Kuntz. Molecular docking using shape descriptors. *J. Comput. Chem.* 13:380–397, 1992.

106 B. K. Shoichet and I. D. Kuntz. Protein docking and complementarity. *J. Mol. Biol.* 221:79–102, 1991.

107 M. J. Sippl. Boltzmann's principle, knowledge-based mean fields and protein folding. an approach to the computational determination of protein structures. *J. Comput. Aided. Mol. Des.* 7:473–501, 1993.

108 Graham R. Smith and Michael J. E. Sternberg. Prediction of protein–protein interactions by docking methods. *Curr. Opin. Struct. Biol.* 12:28–35, 2002.

109 M. Totrov and R. Abagyan. Detailed *ab initio* prediction of lysozyme–antibody complex with 1.6 Å accuracy. *Nat. Struct. Biol.* 1:259–63, 1994.

110 Sandor Vajda and C. J. Camacho. Protein–protein docking: is the glass half-full or half-empty? *Trends Biotechnol.* 22:110–6, 2004.

111 A. D. J. van Dijk, R. Boelens and A. M. J. J. Bonvin. Data-driven docking for the study of biomolecular complexes. *FEBS J.* 272:293–312, 2005.

112 P. H. Walls and M. J. Sternberg. New algorithm to model protein–protein recognition based on surface complementarity. Applications to antibody-antigen docking. *J. Mol. Biol.* 228:277–97, 1992.

113 H. Wang. Grid-search molecular accessible surface algorithm for solving the docking problem. *J. Comput. Chem.* 12:746–50, 1991.

114 J. Wang and E. O. Purisima. Analysis of themodynamic determinants in helix propensities of non-polar amino acids through a novel free-energy calculation. *J. Am. Chem. Soc.* 118:995–1001, 1996.

115 Z. Weng, S. Vajda and C. Delisi. Prediction of protein complexes using empirical free energy functions. *Protein Sci.* 5:614–26, 1996.

116 Shoshana J. Wodak and Raul Mendez. Prediction of protein–protein interactions: the CAPRI experiment, its evaluation and implications. *Curr. Opin. Struct. Biol.* 14:242–9, 2004.

117 L. A. Wolsey. *Integer programming.* Wiley, New York, NY, 1998.

118 Z. Xiang and B. Honig. Extending the accuracy limits of prediction for side-chain conformations. *J. Mol. Biol.* 311:421–30, 2001.

119 C. Zhang, J. Chen and C. DeLisi. Protein–protein recognition: exploring the energy funnels near the binding sites. *Proteins* 34:255–67, 1999.

120 C. Zhang, S. Liu, Q. Zhu and Y. Zhou. A knowledge-based energy function for protein–ligand, protein–protein and protein–DNA complexes. *J. Med. Chem.* 48:2325–35, 2005.

121 C. Zhang, G. Vasmatzis, J. L. Cornette and C. DeLisi. Determination of atomic desolvation energies from the structures of crystallized proteins. *J. Mol. Biol.* 267:707–26, 1997.

122 H. Zhou and Y. Zhou. Distance-scaled, finite ideal-gas reference state improves structure-derived potentials of mean force for structure selection and stability prediction. *Protein Sci.* 11:2714–26, 2002.

123 H. Zhou and Y. Zhou. Stability scale and atomic solvation parameters extracted from 1023 mutation experiments. *Proteins* 49:483–92, 2002.

18
Lead Identification by Virtual Screening

Andreas Kämper, Didier Rognan and Thomas Lengauer

1 Introduction

The identification of new drugs is a research topic of outstanding interest. Due to recent progress in the determination of several complete genome sequences, including the human genome, the structural genomics projects aiming for structure determination of all naturally occurring protein folds, new techniques for target validation and the advances in bioinformatics, our understanding of the nature of many diseases and their causative facts is constantly increasing. These efforts help to achieve the goal to identify novel small molecules interacting with proteins and in this way find new drug targets. In 2000 it was anticipated that the number of potential drug targets would increase 10-fold [47], which was too optimistic from todays view [177]. The number of druggable proteins is more likely to be around 2200–3000 [81, 174]. Of them, around 600–1500 are disease-related and thus are putative drug targets for small-molecule drugs [81].

The availability of new targets calls for effective systematic procedures for finding putative drugs that bind to these targets. The process of searching through a collection of compounds for molecules showing biological activity against a given target is called *lead identification*. This lead identification is a *screening* procedure (Section 1.1) and part of the overall drug discovery process. It can be subdivided into several individual steps (Section 1.2). As a prerequisite for screening, the molecules which are tested against the target, the *screening compounds* (Section 1.3), have to be preprocessed (Section 2). The actual screening can be performed with a variety of methods outlined in Section 3. The results obtained from these methods need to be analyzed and interpreted (Section 4). The final Sections 5 and 6 of this chapter provide recent case studies and critical evaluations of structure-based and ligand-based virtual screening techniques.

Bioinformatics – From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

1.1 Screening Techniques

Until a few decades ago, the search for drugs was a trial-end-error procedure, with the target proteins being mostly unknown. In the last few decades of the 20th century, two different systematic techniques for searching for drugs have become accessible. Both of them are based on the fact that, increasingly, the target proteins for drugs or putative drugs have been identified. The two approaches are:

- High-throughput screening (HTS) is an experimental technique, where in a fully automated fashion a robot tests all molecules from a library against a molecular test system [78].
- Virtual screening (VS), on the other hand, is a pure computational technique. Here, the computer is used to estimate biological activities, e.g. binding affinities. This includes one or more computational techniques.

These techniques can complement each other in the sense that VS guides the experimental setup of HTS, but recently VS is more and more being seen as an alternative to HTS [101]. There are many concepts for the integration of both approaches [7, 69], showing the benefit of including experimental and *in silico* methods in drug discovery. As an example, VS methods can be used to select a subset of compounds for HTS or to analyze the results of a HTS experiment.

Due to their different nature, VS and HTS techniques have different advantages and disadvantages. For HTS, the major drawback is the cost of the experiments. The cost is mainly determined by the purchase of compounds of about US$1.00 per compound [198]. This has to be multiplied by the number of compounds used per HTS run, typically on the order of a few hundred thousands. In addition, supplies and an assay are needed. For both, the cost is highly dependent on the type of target. On the other hand, the major limitation for VS is the need for prerequisite knowledge about the binding process. If there are neither known actives which can serve as templates nor a three-dimensional (3-D) structure of the target protein, VS cannot be used. If the 3-D structure of the target is known, then methods of structure-based design can be used (see Chapter 16). The other possibility is that at least one ligand is known that binds to the target, such as the natural substrate or another inhibitor. In the latter case, methods of ligand-based design can be applied. A substantial advantage of VS is its applicability to not yet synthesized, *virtual* compounds. This facilitates screening of virtual combinatorial libraries with up to billions of molecules. It is obvious that VS methods must be very efficient to deal with such large numbers of compounds. Thus, often not only a single technique is used for VS. Instead, screening proceeds in a sequence of steps, each of which reduces the number of considered compounds, starting with very fast techniques, followed by more advanced, computationally more expensive techniques.

Within this chapter we will cover all computational aspects of VS. The sections on preprocessing of libraries (Section 2) and postprocessing of hit lists (Section 4) are also valid for HTS. However, we will exclude the more technical aspects of data handling and information storage of HTS. Furthermore, the structure-based design techniques covered in detail in Chapter 16 will not be included here.

1.2 Drug Discovery Process

Drug discovery is a time-consuming and expensive process [44] which involves a number of steps. Although the process is not linear – several of the steps have to be repeated iteratively – it is often represented as a pipeline (Figure 1). Within the pipeline, screening is performed during hit identification after a suitable drug target has been identified and validated. A hit can be defined as a compound which exhibits a strong binding affinity to the target. In order to perform a screening run, first a collection of compounds (Section 1.3) is submitted to the pipeline. This collection has to be prepared for screening using a number of preprocessing steps (Section 2.1). Specifically, unsuitable compounds are discarded by filtering steps. Next, within the pipeline a structure- or ligand-based technique is applied to further restrict the number of compounds (for ligand-based methods, see Section 3; for structure-based methods, see Chapter 16). This is not necessarily performed in a single step. It is quite common to use several cascading techniques, starting with a fast but inaccurate method to exclude many compounds, ending with a slow, but better method to screen the most promising compounds (see Sections 5

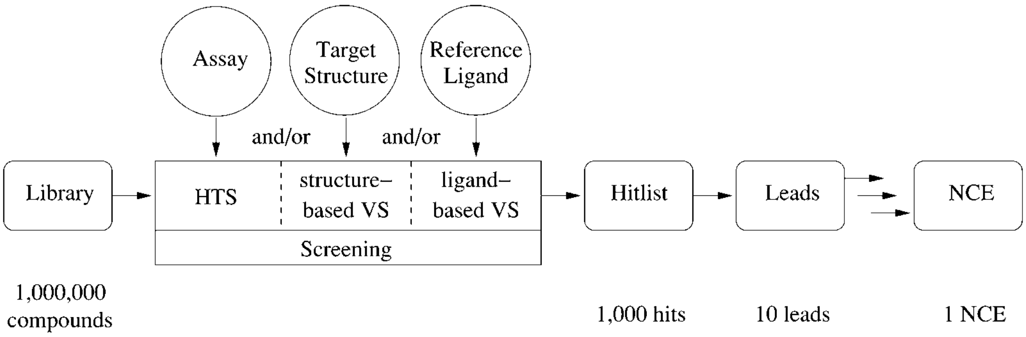


Figure 1 The drug development process. Screening is applied for reducing the number of initial compounds to a hitlist of molecules with a high binding affinity. Compounds from the hitlist are subsequently optimized to leads. The final steps (not shown) are then the finding of candidate structures, clinical trials and, finally, the approval of the new chemical entity (NCE) by the authorities.

and 6 for recent examples). Finally, a number of molecules exhibiting a strong binding affinity to the target are obtained – the *hits*. The crude hitlist obtained by these methods needs to be analyzed and compounds need to be sorted to prioritize subsequent lead selection (Section 4). In further steps of drug development, top-ranking compounds of the hitlist are refined and a small number of lead structures exhibiting promising properties are obtained (*hit-to-lead*) which may be optimized further to finally become *candidates* used in clinical trials. The respective drug optimization techniques are described in Chapter 19 of this volume.

1.3 Compound Collections

The number of all possible drug-sized molecules, the virtual chemistry space, is huge. A systematic exploration of a small part of this space with molecules up to 11 heavy atoms was recently performed [57]. After exclusion of unsuitable chemicals with many small rings, over 13 million different compounds remained. A typical drug molecule can be up to twice as large as the compounds investigated in this study (average mass of 340 Da [55], about 24 heavy atoms). Estimates of the number of "drug-like" molecules accessible to current synthesis procedures are of the order of 10^{60} [18] to 10^{100} [200]. These numbers indicate that even when combining all compounds ever synthesized (estimated 10^{8} molecules), we cover an almost negligible fraction of the virtual chemical space.

Compounds for screening can be obtained from databases of known structures, from combinatorial libraries or from *de novo* design programs. Due to problems with synthesizability, often only known structures are considered. Typical databases with organic laboratory compounds [e.g. MDL Available Chemicals Directory (ACD, http://www.mdli.com/products/experiment/available_chem_dir) or SPRESI (http://www.spresi.com)] are not suitable sources for screening compounds due to the nondrug-like properties of most of the entries. (In fact, these databases are used as references for nondrugs, see below.) Much better sources are collections available in-house to pharmaceutical companies or offered by screening compound vendors, containing historical compounds and combinatorial libraries. Within the MDL Screening Compounds Directory (SCD, http://www.mdli.com/products/experiment/screening_compounds) database, over 3 million screening compounds are listed together with supplier information. Unfortunately, all these compound databases need extensive cleanup to be suitable for drug screening. Very recently, ZINC, a curated large screening library of purchasable compounds has become available [85], in which all the necessary preprocessing steps (Section 2.1) have been performed.

Reference data on pharmaceutical compounds at various stages of development can be taken from the MDL Drug Data Report (MDDR, http://www.mdli.com/products/knowledge/drug_data_report), the World Drug Index (WDI, http://scientific.thomson.com/products/wdi), or the MDL Comprehensive Medicinal Chemistry (CMC, http://www.mdli.com/products/knowledge/medicinal_chem) database.

2 Filtering and Preparation of Ligands

The data from screening compound collections are usually not suitable for VS off the shelf. On the one hand, this is due to incomplete information (often missing 3-D coordinates, stereochemistry, hydrogen atoms). On the other hand, chemical libraries tend to contain a number of undesired compounds and a lot of duplicates. Thus, before a library of compounds can be used in VS, a number of preparatory and filtering steps (Section 2.1) have to be applied. Among these filters, bioavailability (Section 2.2) and drug-likeness (Section 2.3) of the compounds are of special relevance. The overall preparation process is summarized in Figure 2.

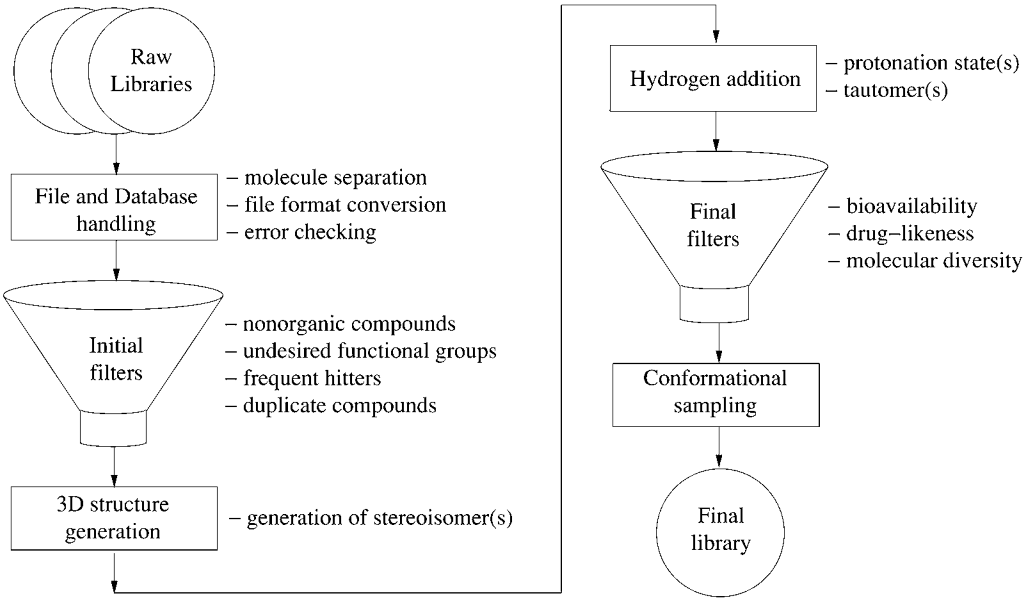


Figure 2 Example preprocessing workflow for chemical libraries (see text for details).

2.1 Library Preprocessing

An initial preprocessing step comprises the separation of entries with more than a single molecule (e.g. containing a charged species and its counterions) into their constituent entries. Next, all nonorganic molecules (e.g. chloride ions, water) must be removed. The most obvious and easiest method is the removal of all molecules without any carbon atom. Alternatively, as almost all drugs contain a bond between carbon and a hetero-atom, another strategy is to remove molecules without any of these bonds.

Then the library is subjected to a functional group filter. Here substructure search is performed in order to identify and discard compounds with known undesired groups in the library. This technique can be applied to reactive functional groups [170], groups unlikely to be leads, promiscuous binders and even functional groups classified as toxic [201]. Strategic pooling, a technique proposed by Hann and coworkers [75], can also be applied by using functional group filters. For instance, if the ligand is required to contain an acidic group, only compounds with this function are integrated into the final library.

A common problem of HTS and VS consists of small molecules binding to many different proteins, resulting in false-positive results (so called "frequent hitters" or "promiscuous binders") [135, 172]. By statistical methods based on substructures, Roche and coworkers [172] developed a scoring scheme based on a neural network to classify molecules as frequent hitters. Merkwirth and coworkers [141] use an ensemble model for this classification. Molecules covalently binding to proteins can be filtered out using reactive functional group filters (see above).

Finding duplicates in screening libraries significantly reduces the number of compounds to be tested. A one-by-one subgraph matching of connection tables for all pairs of compounds is too expensive for collections with more than a few thousand molecules. Instead, a representation of the molecular structure which can be compared easily is generated. Often chemical hash codes like those provided by Ihlenfeldt and Gasteiger [84] are used. Here, the molecular topology is encoded in a single number – the *hash value*. If two hash values are different, the molecules are different. If the hash values are identical, the two compounds are most likely identical. Only in extremely rare cases do two different compounds exhibit the same hash code. To be sure, a substructure matching has to be performed if two identical hash codes have been found. An alternative to hash codes is the generation of a unique string for every molecule. This can be done by using the SMILES (Simplified Molecular Input Line Entry System) representation after Weininger [210]. An enhancement of the SMILES representation can generate unique strings [209] by using an atom-ordering scheme, which is suitable for molecular comparison.

The methods described so far need the 2-D representation of the molecule in form of a connection table only. If 3-D information is needed further in the screening pipeline then, as a next step, this information must be generated from either 2-D coordinates [e.g. most structure data files (SDFs)] or connection tables [e.g. structures in 1-D string representations, like SMILES or Sybyl Line Notation (SLN)]. Due to the still large amount of data this can only be done efficiently by structure generation programs. These convert the connection table into a single low-energy conformation. The two most widely applied tools of this kind are CONCORD [157] and CORINA [176]. CONCORD uses rule sets based on literature values of bond lengths and torsion angles to construct the molecule. Based on these values, acyclic parts of the molecule are constructed. For cyclic parts special rules pertaining to ring geometries are applied. Ring systems are obtained by merging conformations of the individual rings and optimizing the geometry such as to minimize the strain. CORINA works similarly, but includes more literature data and has a backtracking algorithm for generation of strained ring systems.

While the 3-D structure is generated, information on the stereochemistry of the compound has to be included. The often incomplete annotation of the stereocenters is a pervasive problem with current compound collections. Each stereocenter offers a choice of two stereoisomers (for the relevant cases asymmetrical carbons and *cis–trans* isomerism). Which of these alternatives are explored is up to the user. An exhaustive generation of all possible stereoisomers of the compounds is one extreme, discarding the molecule due to incomplete information is the other. Typically, only a small number of stereoisomers are generated at this stage. Some programs even allow to handle stereocenters as variable during the calculation [62].

Next step in preprocessing, although often combined with structure generation, is the addition of missing hydrogens. This includes the assignment of a protonation state and an assessment of the tautomerism of the molecule. Depending on the application, either the most likely or all protonation states/tautomers are generated. A typical approach for assignment is the use of empirical rules. For example, carbonic acids are usually kept deprotonated and primary aliphatic amines are protonated. A program for generation of protonated forms is LigPrep from Schrödinger LLC (http://www.schroedinger.com). Tautomers can be generated with TAUTOMER (http://www.mol-net.de/software/tautomer) by Molecular Networks.

Depending on the methods used for screening, often the conformational space of a molecule needs to be explored by a VS program. While the structure generation programs (see above) produce a single structure only, the conformational analysis programs produce a set of alternative low-energy conformations for a molecule. Leach has reviewed the available techniques [119] and several available programs have been compared [20] whether they predict bioactive conformations correctly. The OMEGA program (http://www.eyesopen.com/products/applications/omega.html) by OpenEye, using a rule-based algorithm, currently seems to provide the best trade-off between accuracy and speed [19, 20].

During all the steps of the screening procedure the identity of the molecules (e.g. their registry numbers, order information) has to be maintained and stored, typically in a relational database system.

2.2 Bioavailability

It is highly desirable for a drug to be administered by oral ingestion in order to be easily utilized by the patient. Thus, the molecule must have reasonable aqueous solubility and has to pass the intestinal membrane in order to enter blood circulation. Bioavailability – a transport phenomenon – is often confused with drug-likeness. Here we keep these two entirely different subjects separate. We discuss bioavailability in this section and drug-likeness in Section 2.3.

By analysis of molecules that have entered clinical trials (and, thus, are bioavailable), Lipinski and coworkers established the "Rule of Five", which provides a simple heuristic rule for oral bioavailability [128]. It is likely that a molecule exhibits poor absorption, if two or more of the following criteria are fulfilled:

- Number of hydrogen bond donors (counted as number of O–H and N–H groups) > 5
- Number of hydrogen bond acceptors (counted as number of any O or N atom) > 10
- Molecular weight > 500
- Calculated $\log P > 5$ (if $C \log P$ is used, see below)

Here, $\log P$ represents the logarithm base 10 of the octanol–water partitioning coefficient, a property which can easily be calculated by property estimation techniques [143]. Among the estimation techniques for $\log P$, $C \log P$ introduced by Hansch and Leo [76] and available from BioByte (http://www.biobyte.com) is widely accepted. The idea of simple property-based rules as rejection criteria for bioavailability was extended by Ghose and coworkers [64]. Here, ranges were calculated for $\log P$, molar refractivity, molecular weight and number of atoms.

After development of fast estimation methods for the polar surface area (PSA) by Clark [31] and Ertl and coworkers [50], rejection of molecules with a PSA $>$ 140 Å was proposed as the only rejection criterion. Veber and coworkers [193] analyzed a database with drug candidates. They classified compounds as bioavailable, if the PSA $<$140 Å and the number of rotatable bonds less than 12. More detail on bioavailability and, more generally, on absorption, dissipation, metabolism and excretion (ADME) properties is provided in Chapter 19.

2.3 Drug-likeness

With the filtering methods described above, the knowledge of medicinal chemists of whether a compound might be a good drug or not is not taken into account. It is desirable that the compounds in a screening library have the typical properties of drugs. Thus, a binary classification, whether a compound is "drug-like" or not, must be performed on all compounds. The challenge regarding this problem is that "drug-likeness" is a property which is not easily evaluated and not related in a simple fashion to other chemical, physical or biological properties. In order to solve the decision problem, the knowledge of medicinal chemists for assessment of "drug-likeness" is used. The implicit knowledge on drug-likeness is inherent in the databases of known drugs and can be extracted by comparison with databases of nondrugs [5, 65, 175].

Implicit information on drugs is contained in the MDDR, WDI and CMC databases (see Section 1.3). Within these databases, not all entries are existing drugs, but the majority of entries were designed by medicinal chemists with the intention of developing a drug. In contrast, databases like the ACD or SPRESI of general organic compounds are supposed to contain very few drugs. A statistical classification technique can be used to extract the knowledge from the databases to decide whether a given compound is a drug or nondrug.

Gillet and Bradshaw [65] used structural features (including number of hydrogen-bond donors and acceptors, number of rotatable bonds, number of aromatic rings, molecular weight) and a shape descriptor. Compounds of the WDI and SPRESI databases were analyzed with a genetic algorithm to derive a weighting scheme, which calculates the drug-likeness of a given compound. Ajay and coworkers [5] used MDL keys (see Section 3.2), in addition. They applied both decision trees and an artificial neural network (ANN) for classification, trained on the MDDR and ACD databases. Sadowski and Kubinyi [175] also used a feed-forward neural network. The classification scheme was trained on atom type descriptors of compounds from the WDI and ACD databases.

Support vector machines (SVM) can also be applied to the drug/nondrug classification problem. In a comparison of SVM to neural networks, Byvatov and coworkers could obtain slightly more accurate classifications on the same data as used in Ref. [175]. Overall, the predictive power of the methods presented so far reaches a typical 80% correctly classified test molecules. Recently, Müller and coworkers [144] were able to reduce the error rate to 7% in a blind-test using SVMs. This performance was achieved by careful model selection after comparison of several learning methods. In summary, the results show that accurate drug-likeness filters can be constructed which use the knowledge on drug-likeness obtained by medicinal chemists over decades.

The technique presented can be tailored to specific screening problems. Thus, it is not the phenomenon of drug-likeness that is assessed, but the likelihood of a drug belonging to a certain class of compounds. Ajay and coworkers [4] used neural networks to classify ligands regarding their CNS activity while Manallack and coworkers [130] used them to screen for candidates binding to kinase and G-protein-coupled receptors. More recently, Briem and Günter [23] presented a SVM method for "kinase-inhibitor likeness". All these target-specific drug-likeness classification techniques have a prediction accuracy of about 80%. Thus, if there is a sufficient number (several hundred) of compounds of a certain class available as training set, machine learning techniques can be trained to predict the likeness to be a certain inhibitor with high accuracy.

2.4 Molecular Diversity

Compound collections, especially those in pharmaceutical companies, contain many series of analogous compounds. It is desirable to screen for only a subset of "maximally diverse" compounds, reducing redundant structural information. This diverse subset is hoped to cover the range of chemical structures and physical properties to a sufficient extent. Unfortunately, there is no generally accepted single definition of similarity or dissimilarity for this purpose [114]. Thus, selecting a set of diverse compounds can be performed in a number of ways. These differ not only in the method, but also in the selection criteria used. These techniques have been reviewed on several occasions [1, 39, 126].

Among the methods for selecting diverse compounds, clustering methods can be considered as the standard technique. For clustering, descriptors (see Section 3.1) are calculated for each compound. Then a cluster analysis algorithm divides a group of compounds into clusters. Compounds within a cluster are similar and compounds from different clusters are dissimilar. After clustering of a library, a representative molecule is taken from each cluster to belong to the diverse compound collection. For clustering of chemical libraries, typically methods generating disjoint clusters are used in which each molecule belongs to a single cluster only. Both, hierarchical [9, 211] and nonhierarchical [213] methods, have been applied. Evaluations of different clustering algorithms demonstrate the better performance of hierarchical clustering methods for several test cases [12, 24, 46]. Naive implementations of hierarchical clustering need an initial $N \cdot N$ similarity matrix, and have space and time complexities of $O(N^2)$ and $O(N^3)$, respectively. By use of the minimum variance method (also called Ward's method) [206] which aims on minimizing the total variance of a cluster, however, a computationally more efficient implementation is possible [146], having space and time complexities of $O(N)$ and $O(N^2)$, respectively. For large numbers of compounds ($N > 10^6$) hierarchical clustering is not applicable. Instead nonhierarchical clustering (e.g. K-means clustering [139]) is used. For these algorithms the time complexity for K generated clusters is $O(KN)$ per iteration for efficient implementations. For descriptions of clustering algorithms, see also Chapters 24 and 27.

A different approach to select diverse compounds is provided by partitioning methods. These use a low-dimensional property space, each property represented by a continuous number that is categorized into a discrete set of value ranges, forming a set of cells in property space [37, 125]. Compounds from a library are then partitioned into these precomputed cells. Two special kinds of descriptors – BCUT descriptors by Pearlman and Smith [155] and pharmacophore keys by Davies [38] – can be used for partitioning chemical property space. BCUT comprises molecular descriptors based on the eigenvalues of a matrix representation of the molecules. These descriptors are designed specifically to define a low-dimensional property space. For the description of the properties, axes are chosen that span property space in such a way that the compounds exhibit maximal variance along the axes and compounds are evenly distributed in property space [155, 156].

Davies [38] developed ChemDiverse, a program using pharmacophore keys for the selection of diverse compound sets. The basic idea is to calculate the pharmacophore key for the first molecule, add the molecule to the diverse compound selection and store the pharmacophore in a list. Subsequently, for the next molecule in the library, the pharmacophore keys are calculated. If this molecule has a pharmacophore key not yet represented in the list, the molecule is added to the selection.

A completely different attempt to find diverse compounds is by dissimilarity-based compound selection. Among them, maxmin by Lajiness [118] is most used. The maxmin algorithm first selects a compound randomly and adds it to the selection. Iteratively, the compound most dissimilar to the already selected set is identified and added to the selection, until a desired number of compounds has been found. A stochastic variant, OptiSim, has been proposed by Clark [32]. The initial random compound is compared to a set of K other randomly chosen compounds. Here, only compounds with a dissimilarity greater than a defined threshold to the already selected compounds are considered. The most dissimilar compound among the *K* compounds is added to the selection. In the next iteration a new set of *K* candidates is generated and the process continues.

Taylor [187] proposed a method based on the stepwise elimination of the most similar molecule from the collection. Initially the similarity matrix between all molecules is calculated. In a stepwise fashion, the two currently most similar compounds are identified, as long as there is more than a single compound left.

While diversity is desired in initial screening runs for identification of hits, once a lead is found, compounds should be similar to the lead, i.e. the library should be "focused". The clustering and partitioning methods can be used directly for generation of focused libraries by switching the selection criteria from one of each kind to all of one kind.

3 Ligand-based VS

The methods of ligand-based VS can be divided into two classes. One class of methods tries to match compounds with identical parts (substructure or pharmacophore) as the active molecule. For these methods, initially, a number of active molecules are analyzed for a common substructure or pharmacophore, which is then used for searching exact matches. The other class tries to find molecules which are "similar" to a known active molecule. The underlying assumption is that if a molecule is structurally similar, it has similar properties, binds in a similar binding mode and exhibits similar activity. This assumption is known as the "similar property principle" [40, 87] or "neighborhood behavior" [153]. Methods for similarity search are applicable if only a single active compound is known. In contrast to substructure and pharmacophore searches, the compounds are not only partitioned with regard to whether they are matching the query or not. Instead, a complete ranking of compounds according to their similarity scores is obtained. Similarity, like dissimilarity, is not clearly defined [114] and there are some remarkable exceptions from the similar property principle [114, 131]. Nevertheless, similarity search is the most widely used method in VS and numerous similarity measures have been developed [181]. Figure 3 illustrates the most common similarity search techniques, detailed below.

The actual methods for defining molecular similarity in this context are quite diverse. Often these methods are classified as 1-, 2- or 3-D, depending on the type of molecular representation used. In this section first we focus on those methods that use the information of a single reference molecule. Then techniques that need a set of input molecules are discussed. The latter are also

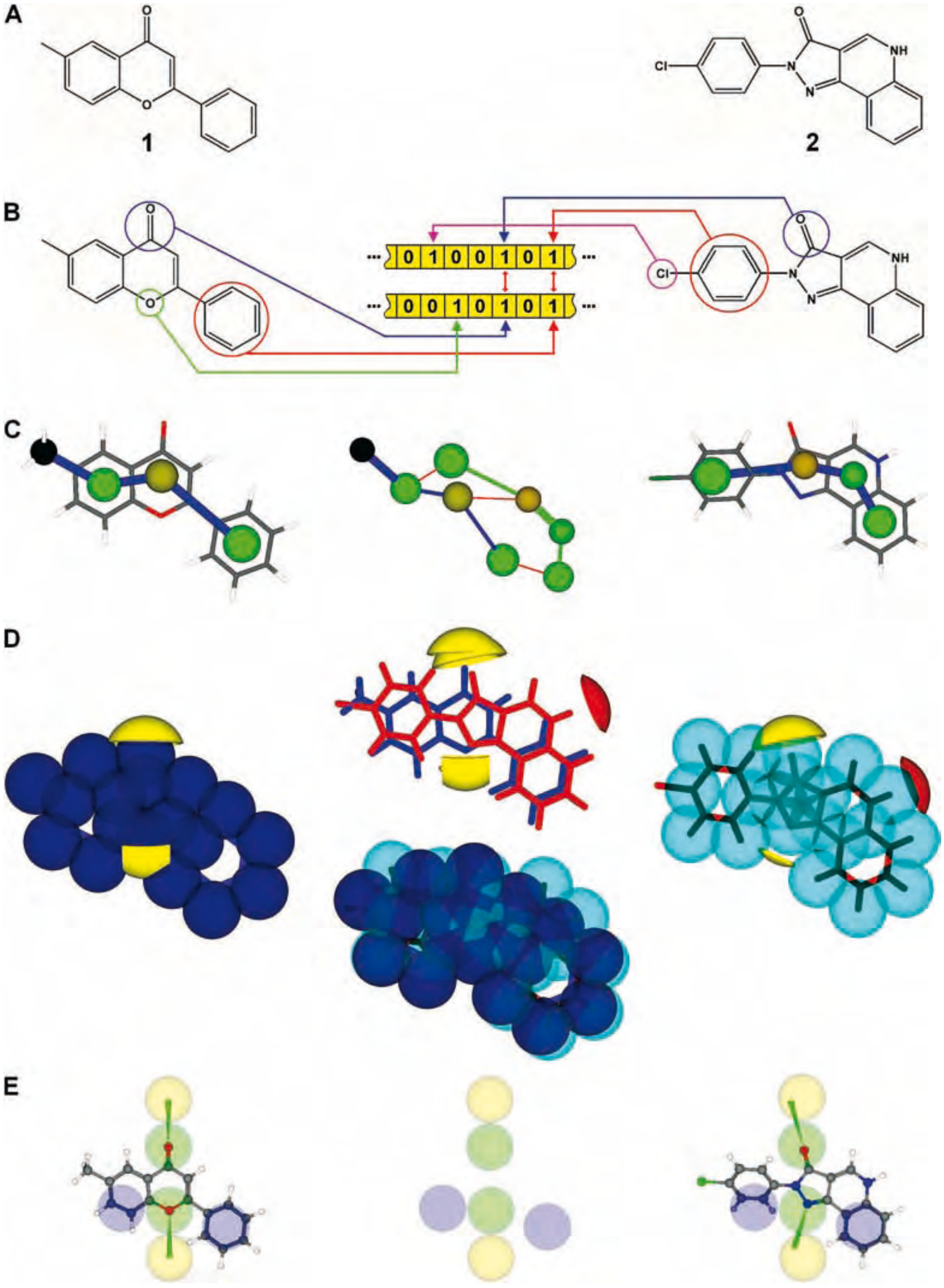


Figure 3 Comparison of common ligand-based screening techniques (see text for details). (A) Example molecules: flavonoid molecule (**1**, test molecule) and a known binder (CGS-9896, **2**, reference molecule) for the benzodiazepine site of the GABA$_A$ receptor [89]. (B) Bitstring generation and comparison. (C) Generation and comparison of feature trees. (D) Comparison by molecular superimposition using FlexS (center top: molecular interaction surfaces, center bottom: molecular volume represented by Gaussian functions. (E) Comparison of both molecules to a pharmacophore model (center).

often used for the selection of compounds from hitlists. These techniques are described in Section 4.

3.1 Descriptor-based Similarity Measures

Similarity search methods have been used for a long time. The field started with counting the numbers of substructures common to a pair of molecules [27,212]. This figure provides an initial effective way of quantifying similarity between molecules with very low computational cost. Since then, it has become a standard retrieval technique for chemical databases.

Methods for calculating quantitative measures for the similarity between a reference molecule and a set of molecules have been studied in detail (see Ref. [211] and references therein). Common to all these techniques is the requirement to provide a set of attributes of the molecules being compared and a similarity coefficient, to provide a quantitative numerical measure for similarity between the molecules. The individual importance of different attributes (e.g. log P, molecular mass, presence of functional groups, etc.) has to be accounted for by definition of a weighting scheme.

Many different similarity coefficients have been proposed in the literature. Some are measures of dissimilarity, while others measure similarity directly. In this chapter, we focus on the most widely used similarity and distance coefficients only, more details can be found in another review [8]. One of the most frequently used distance measures is the Euclidean distance $D_{A,B}^{\text{Euclidean}}$ between two molecules A and B described by properties $x_1 ... x_n$. For continuous property variables, the distance is defined by:

$$D_{A,B}^{\text{Euclidean}} = \sqrt{\sum_{i=1}^{n} \left(x_{i_A} - x_{i_B}\right)^2}. \tag{1}$$

The most often used similarity measure is the Tanimoto coefficient $S_{A,B}^{\text{Tanimoto}}$. This coefficient can be interpreted as the fraction of the number of features present in both molecules divided by the number of features present in at least one of the compounds. For continuous variables it is defined by:

$$S_{A,B}^{\text{Tanimoto}} = \frac{\sum_{i=1}^{n} x_{i_A} x_{i_B}}{\sum_{i=1}^{n} x_{i_A}^2 + \sum_{i=1}^{n} x_{i_B}^2 - \sum_{i=1}^{n} x_{i_A} x_{i_B}}. \tag{2}$$

Willett and coworkers [212] assessed the performance of several distance and similarity coefficients for predicting a measured activity value. The Tanimoto coefficient performed best and, since then, it has become the "standard" coefficient for chemical similarity comparison. Furthermore, this coefficient was shown to be the most appropriate for similarity searches in 2-D databases [212].

In order to compare two molecules in a quantitative fashion, numerical values of attributes of the molecules are needed. The term "molecular descriptor" subsummarizes all numerical representations of chemical information about molecules obtained by a defined mathematical procedure. As example, the molecular descriptor "molecular weight" (*MW*) is defined exactly as the sum of all atomic weights of a molecule:

$$MW = \sum_{i=1}^{n_{\text{atoms}}} m_i. \tag{3}$$

The number of different chemical descriptors which have been proposed for the field of quantitative structure–activity relationships (QSARs) and VS is large (several thousands are described in detail in the handbook by Todeschini and Consonni [189]) and an extensive discussion is out of the scope of this chapter. However, some descriptors have become very important in VS. Among them, $\log P$, *MW* and molar refractivity are typically used in the preprocessing of libraries (see Section 2.2). BCUT descriptors are useful in diversity analysis (see Section 2.4). A second important class of descriptors are bit strings, used in molecular similarity search, as detailed in the next section.

3.2 Bit String Descriptors

For VS the most popular descriptors are based on binary vectors (also called bit strings). The idea of using binary vector representations has its origin in chemical database systems. Bit strings are used in substructure queries to efficiently discard large fractions of the database, before subsequently a much slower subgraph isomorphism algorithm is used. The length of the bit strings varies from roughly 100 bits [218–220] to several million bits [136], depending on the type of information stored. The two most widely used approaches are substructure keys and hashed fingerprints.

Substructure keys [MDL keys available from Elsevier MDL (http://www.mdli.com) and BCI structure fingerprints available from Barnard Chemical Information Ltd. (http://www.bci.gb.com)] represent a description of the substructures present in a molecule (Figure 3B). Within the binary vector, each of the positions is 1, if the corresponding substructure is present in the molecule, 0 otherwise. All substructures for the bit string are predefined in a fragment dictionary. Entries can encode the presence of certain atoms (e.g. there is at least one N present in the molecule) or common functional groups (e.g. ester function). Furthermore, electronic (e.g. *O* with double-bond) and structural features (e.g. six-membered ring) are described by bits.

In hashed fingerprints [Daylight Chemical Information Systems (http://www.daylight.com)], substructure information is encoded by an algorithm. All possible paths of atoms and bonds through the 2-D formula of the molecule, up to a certain certain path length are generated by systematic search, e.g. for a path length of 3 the path C=C–N–C. These patterns are then converted to integer numbers and hashed. The hash code is generated using these numbers as seeds for a pseudo-random number generator, resulting in the fingerprint of this path. The advantage of hashed fingerprints is their applicability to any type of structure without the need of a precomputed fragment dictionary. A disadvantage is the possibility of the same bit string being calculated from different atom paths which may result in false positives, since these hash conflicts are not solved. Unity available from Tripos (http://www.tripos.com) uses a combined approach of substructure keys and hashed fingerprints.

For the use with binary variables, the two similarity coefficients from above can be reformulated. If a denotes the number of bits set (to 1) in molecule A, b for molecule B and c the number of bits set in both molecules, the Euclidean distance becomes:

$$D_{A,B}^{\text{Euclidean}} = \sqrt{a + b - 2c}\,, \tag{4}$$

and the Tanimoto coefficient is given by:

$$S_{A,B}^{\text{Tanimoto}} = \frac{c}{a + b - c}. \tag{5}$$

The substantial advantage of linear descriptions for chemical structures is their speed. The counting of numbers of bits set (a, b, c) can be done computationally very fast, resulting in several hundred thousand molecule comparisons per minute. This allows for fast comparison of millions of compounds in minutes to hours. A disadvantage of the methods described so far is that they cannot detect the similarity between two compounds that behave similar with respect to binding to the target protein, but are structurally quite different [35]. Thus, the techniques cannot find a scaffold different from the scaffold of the reference molecule (scaffold-hopping).

3.3 Feature Trees

Feature trees [166] comprise a class of descriptors in between the classical linear descriptors described in Section 3.1 and molecular superimposition techniques (Section 3.4). In this technique (Figure 3C), a molecule is described as a tree that represents its overall topology. Nodes of the tree represent fragments of the molecule. The nodes are connected by edges if the fragments are also connected by covalent bonds or sharing of atoms. A set of features is assigned to each of the nodes, representing physicochemical properties of the respective fragment. Steric features comprise the number of atoms in the fragment and the approximated van der Waals volume. Chemical features include the interaction profile of the fragment, i.e. whether it acts as hydrogen donor or acceptor, is an aromatic ring, and also represent the hydrophobicity of the fragment.

Similarity between molecules is calculated by matching the two corresponding trees, while preserving their topology. A similarity score quantifies the quality of the fit. The advantage of feature trees provides a more accurate description of chemical properties than by linear descriptors. However, the tree-matching procedure is slower, due to the higher computational complexity of the tree comparison. Typically, thousands of molecules can be compared per minute.

3.4 Molecular Superimposition Approaches

Molecular superimposition techniques structurally align a compound to a reference ligand in 3-D space. During the alignment, matching parts of both molecules are placed on top of each other. The large variety of algorithms for molecular superimposition has been reviewed by Lemmen and coworkers [123]. The application of superposition techniques for VS has also been reported [121].

In general, the molecular superimposition can be achieved in two ways. Either a field-based approach is used, in which properties of the molecules are projected onto a common surface or into three-space. The other method aligns pairs of atoms directly. Early approaches achieved this goal by rigid-body superimposition. Newer programs can handle one or both molecules as flexible on the fly. Nevertheless, the rigid-body techniques are much faster and are thus often preferred. An intermediate technique is to address molecular flexibility by considering a set of alternative conformations of the molecule.

A rigid-body superimposition program reads a reference ligand and a test ligand, then performs an optimization of the position and orientation of the test ligand in space. Early attempts using combinatorial approaches to enumerate efficiently possible matches (correspondences) of chemical features of the two molecules [91, 133] are too computationally intensive. With the program SEAL (Steric and Electrostatic ALignment) [92] and later enhancements [98], for the first time Gaussian functions were used for describing the physicochemical properties of the molecules and a new algorithm was applied to tackle the rigid superpositioning problem efficiently. The description of chemical features by Gaussian functions has several advantages [68]. First, a Fourier transform of a Gaussian is again a Gaussian. Second, there is no boundary which helps in the initial steps of alignment. Furthermore, derivatives can be calculated easily (even symbolically) and, finally, the overlap between two Gaussians increases when their maxima approach each other.

An improvement of the search algorithm was proposed by Lemmen and coworkers [120] in their program RigFit. The optimization is split into two independent optimizations for rotation and translation. Thus, one 6-D search is separated into a sequence of two 3-D searches.

Current state of the art are programs for flexible superpositioning of two molecules. Sheridan and coworkers [180] use distance geometry for superposition, while Itai and coworkers [86] proposed a technique, in which all possible matchings between pharmacophoric points are evaluated in a combinatorial matching. For both techniques, the definition of the pharmacophore is still needed as prerequisite. The combined Monte Carlo and energy-minimization-based technique by McMartin and Bohacek [137] also needs manual intervention. The program GASP [88] was the first method available that was able to handle the structural flexibility of both molecules involved and was not constrained by predefined relationships between functional groups assumed to be similar. GASP is based on a genetic algorithm which mimics the process of evolution. The conformations of each molecule and the correspondences between intramolecular features are coded via so-called chromosomes. In order to modify the superimposition, the chromosomes are subjected to the operations of mutation (local changes) and crossover (splitting and merging of two chromosomes). A population of chromosomes is repeatedly subjected to these modifications and then evaluated with a fitness function. Only the fittest chromosomes survive to the next round. The fitness function used in the selection process of each superposition is calculated by volume overlay, intermolecular matching energy and the conformational energy.

A different technique for superposition, FlexS, uses incremental construction [122]. Here the reference molecule is handled as rigid and the test molecule is flexible. The test molecule is partitioned into fragments which are connected by rotatable bonds. A number of relatively rigid fragments is selected and aligned to the reference molecule. Then the next fragment is attached to the previously placed fragment in all allowed torsion angles. The list of admissible torsion angles is derived by statistical analysis [100] of the Cambridge Structural Database (CSD) [6]. All generated placements are scored by paired intermolecular interactions and overlap, the latter being described by Gaussian functions (see Figure 3D). The best partial solutions are subjected to the next incremental construction cycle until the complete test molecule is build up. The mean computing time is in the order of 30 s per superpositioning for typical test cases.

Krämer and coworkers [107] developed fFLASH, using a fragmentation-reassembly approach. The tool is based on earlier work on FLASHFLOOD [164]. fFLASH describes the query molecule as rigid and the test molecules are handled flexibly. All test molecules are partitioned into fragments by severing rotatable bonds, expanded to a set of conformations, and all conformers stored in a database. Pairs of adjacent fragments are joined and a set of conformations of the fragment pair is generated by varying the dihedral angle at the connecting bond. Molecular interaction features are then calculated and stored in a lookup table. By use of a clique detection algorithm, patterns of features of fragment pairs of the test molecule are geometrically matched on the reference molecule. These matches are subsequently joined, based on the pairwise compatibility of two matches, by a graph algorithm.

3.5 Pharmacophore Searches

A pharmacophore is usually defined as a set of molecular features and their rigid spatial arrangement, which is necessary for ligand–receptor binding [73]. A pharmacophore is typically composed by three to four pharmacophoric centers and their respective distances (Figure 3E). Pharmacophores are applied in 3-D database searches after they have been determined from a set of active ligands. In VS, pharmacophores can also be used as constraints in structure-based screening. Pharmacophores can also act as 3-D descriptors. Often pharmacophores are encoded in the form of bitstrings, known as pharmacophore fingerprints, which are directly applicable for screening (see Section 3.2).

In ligand-based VS, the true pharmacophore is unknown and must be determined first. Pharmacophore perception is closely related to molecular superpositioning, e.g. the program GASP [88] can perform both tasks. Nevertheless, since the programs of both groups are tailored to their specific research areas they are described separately. For automatic determination of a pharmacophore hypothesis a set of active ligands as a training set is needed. The pharmacophore perception is then performed in a number of steps: First, 3-D structures of the molecules must be generated with one of the methods described in Section 2.1. Then the molecules are analyzed in order to identify atoms that can interact with a protein in a characteristic way. Commonly, these pharmacophoric features are acidic and basic groups, hydrogen acceptor and donor sites, aromatic, and hydrophobic groups. In the next step, conformations of the molecules are passed to the pharmacophore perception algorithm. Here, conformations of the molecules are compared in order to identify pharmacophoric features common to all molecules. A number of programs have been developed for pharmacophore identification [73], among them the commercially available tools Catalyst/HipHop [33], DISCO [132] and GASP [88]. For a recent review and a comparison of these three, see Ref. [152]. These programs differ with respect to how the conformations are handled and how the molecules are aligned and compared. GASP uses a genetic algorithm (see Section 3.4) to describe the molecules as flexible. DISCO uses a set of low-energy conformations which are kept rigid throughout the calculation and a clique-detection algorithm is used for rigid-body alignment. Catalyst/HipHop also uses a set of rigid low-energy conformations of the molecules, but then performs a pruned exhaustive search to identify configurations common to all molecules. Once the pharmacophore is identified, it can be used to screen a 3-D database.

An extension of the ligand-based pharmacophores described so far is the use of structural information of the receptor for pharmacophore generation. Two recent examples of these structure-based pharmacophores are the works of Wolber and Langer [214] and Griffith and coworkers [72].

3.6 QSARs

The structure and the physicochemical properties of a molecule can be used to model its biological activity. The mathematical description of this relationship in a quantitative way is the aim of QSAR techniques. In order to model structure–activity relationships, first, for each compound in the library a number of molecular descriptors have to be calculated. In a second step, a quantitative relationship between these descriptors and the activity is derived. This section covers only some selected techniques from the field of QSAR, which has grown in terms of using more and more sophisticated descriptors, and also more sophisticated statistical tools for finding correlations between structure and activity.

The classical technique is Hansch analysis which correlates activity with physicochemical properties by use of regression analysis. Hansch and coworkers [77] described the dependency of the concentration C needed for a certain biological response in terms of the hydrophobicity (expressed by the $\log P$ value) and electronic effects (using the Hammett constant σ) by the equation:

$$\log \frac{1}{C} = k_1 \cdot \log P + k_2 \cdot \sigma + k_3. \tag{6}$$

Here, the k_i are the coefficients to be fitted by the regression. Using this type of QSAR analysis, today, several thousand successful applications have been reported and a database of QSAR equations is electronically available (http://www.cqsar.com/medchem/chem/qsar-db). The descriptors applied include steric, electronic and hydrophobic effects as well as indicator variables. These values are obtained either by computer prediction techniques or experimentally. Due to the large number of descriptors available, the dependency between them has to be studied in order to find the relevant ones. This "feature selection" is usually done based on principal components analysis (PCA) [61] or its extension partial least squares analysis (PLS) [82].

An extension of the classic approach was the introduction of 3-D information of the ligands to reflect the geometry of their binding to receptors, including their chirality. The first of this 3-D-QSAR techniques was the Comparative Molecular Field Analysis (CoMFA) [34] which turned out to be very successful (many examples can be found in Refs. [111–113]). In CoMFA a set of molecules is selected which have an identical binding mode, i.e., they bind to the same site in the same relative geometry. To derive the CoMFA model, for all training set molecules, first, partial charges are assigned and low-energy conformations are generated. Then the molecules are aligned by use of a pharmacophore hypothesis and positioned inside a 3-D grid. For each grid point and for each molecule separately, "field" values (interaction energies) are calculated for charged and uncharged probe atoms. Finally, PLS analysis is used to correlate the fields with biological activity data. The result of this analysis is typically represented as a set of contour maps showing favorable and unfavorable regions for certain substituents. Several techniques have been proposed to obtain better fields. By calculating fields with GRID [71] or HINT [94] more different probes can be used which allows modeling of a wider range of interactions. By replacing CoMFA potentials with SEAL (see Section 3.4) similarity fields (Comparative Molecular Similarity Indices Analysis, CoMSIA [99]) the results become more stable. A frequent problem for PLS can be the high number of noise variables not contributing to the description. With GOLPE (Generating Optimal Linear PLS Estimations) [10] the meaningful variables can be selected and the predictive ability of the model is checked by cross-validation. A cause of error for CoMFA, CoMSIA, GRID/GOLPE and related techniques is the mutual alignment of all molecules. There are methods available that retain the 3-D information, but are independent of the alignment. Examples for these techniques are WHIM (Weighted Holistic Invariant Molecular indices) [188], which uses the moments of atomic properties as descriptors, and the related technique MS-WHIM [21], which uses molecular surface points instead of the atoms as descriptors.

In the classical 3-D-QSAR methods described above, only information on the geometry of the ligands is used. In 4-D techniques multiple conformations or orientations of the ligands are considered simultaneously. It is even possible to include information on the protein structure to which the ligands are bound in the QSAR model. The program Quasar by Vedani and coworkers [195] is a method which constructs a receptor-surface model and bridges between 3-D-QSAR and receptor modeling, taking induced fit into account. Currently, multidimensional QSAR studies are extended up to six dimensions to allow for the simultaneous consideration of different solvation models [194].

3.7 Other Techniques

The interaction of a ligand with a target molecule can be described in terms of the respective molecular surfaces, that have to be complementary with respect to both physicochemical properties and shape. Finding optimal surface complementarity is the main aim of docking procedures (see Chapter 16). Thus, the comparison of different ligands in terms of their molecular surfaces and the properties mapped to them is a valuable similarity criterion.

Among the many techniques of molecular surface comparison, we focus on the recent graph-based method SURFCOMP of Hofbauer and coworkers [79] and the gnomonic projection method [17] as examples. The comparison of two surfaces, each described by a point set in three-space is not an easy task. The problem can only be solved efficiently if the surface model is simplified. In SUFRCOMP, first a representation of the surface via overlapping circular patches is calculated. Then the centers of these patches, representing critical points, are reduced in number using a number of filters and matched via maximal common subgraph comparison.

Blaney and coworkers [17] use gnomonic projection of the molecular surface properties onto equispaced points on the surface of an enclosing sphere. To do so, the points in space at which vectors from the sphere's surface to the "center" of the molecule cut the molecular surface are calculated. The physicochemical properties on the cutpoint farthest from the "center" are then projected onto the sphere. The comparison of the projections of two molecules is then performed after mapping of the property values on two dimensions.

A different type of measuring similarity between molecules is the use of "virtual affinity fingerprints" [124, 208]. In the Flexsim-X method by Lessel and Briem [124] ligands are flexibly docked into a carefully selected reference set of protein-binding sites using the FlexX docking program [165]. The highest-ranking solution of each docking run is selected. The virtual affinity fingerprint of a ligand is then defined as the vector of docking scores obtained for the different binding pockets. Molecules are compared by the Euclidean distance between their affinity fingerprints. The technique was shown to detect molecules with similar biological affinity without prior knowledge of the target protein structure. An extension of this work to calculate similarities of functional groups is Flexsim-R [208].

4 Postprocessing of Hitlists

HTS or VS runs of a compound collection with up to millions of entries results in a huge volume of data. The obtained list of hits is rather crude and needs substantial clean-up. There are a number of computational methods for the postprocessing and analysis of screening data. First, the output is simplified by removing data points where the screening failed (e.g. no docking solution, failure during experiment). Here, also data not needed in postprocessing (e.g. intermediate results, log files) are discarded. Second, the most promising hits have to be selected mostly on the basis of their rank in the hitlists or by criteria based on the scores, in order to reduce the data set to manageable size. To identify leads among the screening data is a challenging problem, addressed by a number of different computational methods. The often concealed information can be extracted by data-mining procedures (Section 4.1). A general problem of screening data consists of false-positive results. Especially for results of structure-based VS runs, some techniques have been developed to identify and discard false positives (Section 4.2). Whenever a combination of different techniques is used in screening, each technique results in a different hitlist. Here, consensus techniques help in picking hits (Section 4.3). Nevertheless, the most important method is still the visual inspection of the results by an experienced medicinal chemist, assisted by visualization tools (Section 4.4).

4.1 Data Mining

A common approach for mining hitlists is the search for families with similar chemical structure among the active compounds. Here active compounds (actives) are those with high affinity, high scores or high similarity after screening. Chemical families can be identified by grouping the compounds with similar chemical structure. A chemical family is characterized by a common scaffold. Substructure search among the results can be applied to identify these families. Roberts and coworkers developed LeadScope [171], a structural classification technique. The method classifies compounds into a collection of predefined chemical families. The predefined families are arranged hierarchically, starting with a major structural class on top, which is subdivided further. For example, a 3-methoxy-pyridine derivative is found in the pyridine → pyridine, 3-R → pyridine, 3-alkoxy class of the hierarchy. For each structural class, activity data and frequency in the data set is depicted in an intuitive bar plot.

As an alternative, techniques for similarity search (Section 3) can be applied to identify families. In this case, the families are defined by a high degree of similarity. For grouping the families, clustering techniques are often used (as described in Section 2.4). Due to the importance of hitlist mining, a number of dedicated clustering techniques have been developed [49, 185].

Another approach to data mining using classification techniques is recursive partitioning (RP) [173,221]. RP is a nonparametric classification technique (as opposed to the many parameters in QSAR models), in which the whole set of compounds is recursively classified into disjoint subsets using statistically determined rules. In this manner, a tree is constructed, in which some terminal nodes (leaves) are enriched with actives, while other leaves contain mostly inactive molecules. If the path from a leaf with actives is traced back to the root node, the molecular descriptors used for partitioning at the inner nodes can be used to characterize or to search for actives.

Nicolaou and coworkers developed a classification method using a phylogenetic-like tree (PGLT) [147]. This tree is constructed using a combination of techniques. Each node has bins for active and inactive compounds. First, all active molecules are stored in the active bin of the tree's root node. Then, in an iterative fashion, a clustering of the molecules of the current leaf is performed, using a criterion based on chemical descriptors. In a next step, cluster level selection is performed to select a set of "natural" clusters. Each of the "natural" clusters is then subjected to a maximum common subgraph (MCS) search. Common substructures are evaluated by a set of rules to evaluate each and to discard all those not providing new knowledge. The rules, for example, discard substructures already found in other nodes, or those identical or subsets of the parent node. Then, for each of the remaining substructures, all molecules from the parent node containing the respective MCS are added to a newly created tree node. Finally, a node is selected at which the iteration proceeds. After the actives have been used to construct the tree, a postprocessing procedure is performed in order to prune the tree and reduce it to contain only nodes with structurally homogeneous families. This is done by adding inactive compounds to the inactive bins of the PGLT using the substructure rules derived with the actives. For each node, the similarity between actives and inactives is calculated and nodes with dissimilarities are eliminated. The technique described has been implemented in the program ClassPharmer [Bioreason (http://www.bioreason.com)].

4.2 Analysis of the Protein–Ligand Interface

A particularly interesting type of strategy that can be applied to results of structure-based screening is the analysis of structural properties of the bound protein–ligand complex. Although this method also belongs to docking techniques (see Chapter 16), we describe it here as a representative example for an important class of postprocessing techniques. Current scoring functions favor the formation of many protein–ligand hydrogen bonds and salt bridges, even if the structures exhibit only limited steric complementarity overall due to holes along the interface or larger parts of the ligand being exposed to the solvent. Stahl and Böhm [184] propose a postprocessing procedure of docking results. For a set of generated docking poses, first, all poses with close contacts between polar atoms that do not take part in hydrogen bonds are discarded. Then the fraction of ligand volume located inside the cavity is calculated. Poses with less than average buried volume are discarded. The size of lipophilic cavities at the protein–ligand interface also acts as filter criterion: poses exceeding the minimum value by more than 25 Å are discarded. Finally, the solvent-accessible surface of nonpolar parts of the ligand is calculated and used for rescoring.

Giordanetto and coworkers [67] also propose the use of solvent-accessible surface areas. These authors perform a classification of all receptor and ligand atoms into classes, depending on the physicochemical properties of hydrophilicity, charge and hybridization. Then descriptors are calculated that describe the energetic cost of burying the atoms. In addition, conformational entropy differences between holo and apo form of the protein are calculated. Here, an amino acid-based conformational entropy contribution of the protein after Murphy and Freire [145] to the binding affinity is used. By use of these techniques, affinity predictions could be improved on the cost of less accurate binding mode prediction.

Results from docking studies can also be analyzed by structural interaction fingerprints as proposed by Deng and coworkers [41]. These interaction fingerprints are a translation of the structural information of a protein–ligand complex into a binary vector. The technique can be applied for identification and clustering of similar docking poses.

4.3 Consensus Techniques

The combination of several different computational methods is another approach to reducing the number of false positives and prioritizing molecules for further study. Some of these methods are only applicable to structure-based techniques, while some use mixtures of different computational methods, including ligand-based techniques. The prototype of the structure-based methods in this field is consensus scoring [28]. Here, one docking program is used to generate a docking pose. Then the highest ranking structure is re-evaluated with different scoring functions. If the compound is not among the top-scoring compounds for all scoring functions applied it is discarded. In a computer experiment by Wang and Wang [205] it has been shown that hit rates improve significantly after consensus scoring if three or four scoring functions are used.

Methods using not only different scoring functions, but different docking techniques go a step further [154]. In the ConsDock approach, docking is performed with three different docking programs and a set of 30 top-ranking poses is stored obtained with each of them. Then a hierarchical clustering is performed on each set and the highest-ranking pose within each cluster is defined as its "leader". Consensus pairs are defined, where two of the docking programs result in similar leaders. Each of these pairs is then described by its mean and clustered again into classes. Finally, the mean pose of the clusters is subjected to re-ranking according to the number of entries in each class.

The use of entirely different computational techniques for investigation of hitlists has been proposed by some groups. Klon and coworkers [102,103] use a combination of docking and machine learning. First, docking of a library is performed with three different docking programs. Then a naive Bayesian classifier is trained on the docking scores of the top-scoring compounds, which are labeled as "good", if their score is better than a threshold. The compounds themselves are described by an extended-connectivity fingerprint as structural descriptor [Pipeline Pilot program available from SciTegic (http://www.scitegic.com)]. Application of the Bayesian classifier for re-ranking the hitlists improved the enrichment in most of the test cases, without any *a priori* knowledge of the activity of the compounds.

Especially in docking, the high-dimensional search space can be explored a bit further to re-rank hitlists. On the one hand, a multiconformer description of the protein can be used [199]. On the other hand, not only the top-ranking pose, but several poses can be used for calculating the score [104].

Ginn and coworkers [66] proposed the use of data fusion for combining molecular similarity measures. In this procedure, a similarity search is performed with at least two different similarity measures i. The rank positions r_i of each individual structure in the hit lists are then combined to a new score. With the fusion rule $\sum_{i=1}^{n} r_i$ the performance is at least as good as the best individual measure.

4.4 Visualization

For the simultaneous display of screening-result data in several dimensions, a number of techniques are available [3, 63, 116]. The techniques have been implemented in several tools for display of screening data using highly sophisticated graphical data representations for visual data mining [DecisionSite (Spotfire; http://www.spotfire.com), ClassPharmer (Bioreason; http://www.bioreason.com), LeadNavigator (LION Bioscience; http://www.lionbioscience.com)]. Results are plotted in multiple dimensions, combining data from different databases. The data points in the plots are linked to the corresponding chemical structures and vice versa. This enables the medicinal chemists to identify patterns within the results. A technique for visualization of the multidimensional screening data is the nonlinear mapping of the data to a lower-dimensional space with just two or three dimensions. The usual technique for nonlinear mapping is multidimensional scaling [110]. This technique aims at keeping points close together in low-dimensional space if they are also close together in the original data-space.

With recent enhancements [2, 216], multidimensional scaling is applicable to large-screening data sets. Despite all the efforts in visualization techniques, it has been pointed out that visual data-mining tools are not applicable to extremely large and complex data sets [147]. Furthermore, due to their "interactive" approach, these tools cannot readily be integrated into fully automated screening procedures.

5 Critical Evaluation of Structure-based VS

Nowadays a large collection of docking/scoring tools is available for high-throughput virtual screening. Out of the flow of information generated over the last 5 years, a computational chemist entering into a VS project will have to make a few decisions about the screening strategy and the tools which are the most suited to their project. Section 5.1 is aimed at pinpointing some good practices in order to avoid classical failures. Section 5.2 will review some recent success stories which could inspire the reader for future work.

5.1 Influence of Parameter Settings

Several input parameters may affect the effectiveness of a VS run. Depending on the computational tool that has been chosen, the number of parameters may vary from around a dozen to over 100. It is therefore crucial to select the best possible input settings, which unfortunately are not always known in advance. However, a few robust guides based on current knowledge can be derived.

5.1.1 Which Library?

As reported above (Section 1.3), several commercially available compound collections are available. There is usually no reason to favor one particular compound collection over another. As most of them are easily accessible [11, 182], the best possible approach for an academic user is to start from a unified and filtered data set [85]. Of course, corporate and focussed/targeted libraries may also be used. They are particularly interesting for screening targets belonging to heavily investigated families (e.g. kinases, GPCRs) and containing a high percentage of true positives.

Whatever the database selected, it is generally advisable to downsize the number of molecules which will be submitted to 3-D docking. Apart from some important filters [chemical reactivity (see Section 2.1), drug-likeness (see Section 2.3), etc.], it is important to remove molecules which do not fulfill simple 2- or/and 3-D pharmacophoric features. This simple strategy aids in dramatically reducing the number of potentially interesting compounds without losing many true positives [51, 129]. If one is simply interested in setting-up optimal screening conditions (e.g. discriminating a few true actives from randomly chosen decoys), it remains important to carefully set-up the test data set in order to avoid artificial enrichments in true actives by making sure that chemical spaces covered by actives and inactives/random compounds largely overlap [197].

5.1.2 Which Ligand Conformation(s)?

Most docking programs only require a single low-energy conformation for each ligand of the data set, provided by automated 3-D converting utilities [157, 176]. For docking tools requiring a multi conformer ligand library, it is important to start from biologically relevant conformations. Several studies agree to conclude that the most reliable conformations are not necessarily produced by the most accurate and CPU-demanding methods. A safe start is to use fast conformer generators like Omega (http://www.eyesopen.com/products/applications/omega.html) or Catalyst [33], which accurately sample the biologically relevant conformational space for a wide array of chemotypes [19, 70, 93, 96].

5.1.3 Which Protein Coordinates?

When screening a high-resolution X-ray structure, several input coordinates might be available describing either ligand-bound (holo) or a ligand-free (apo) structures. A systematic survey over nine enzymes unambiguously demonstrates that the holo form, if it exists, should be the first choice [19]. Furthermore, X-ray structures appear to clearly outperform the corresponding homology models in discriminating known inhibitors from random decoys [134, 150]. However, if the sequence identity (on binding site-lining residues) to the X-ray template is higher than 50%, comparable enrichment rates in true inhibitors can be found [150]. This encouraging result suggests that genomic-scale VS might be feasible, provided that an accurate description of the binding sites can be drawn from existing X-ray templates.

5.1.4 Which Docking Tool?

Starting from the pioneering work of Kuntz and coworkers [115], numerous docking programs based on very different physicochemical approximations have been reported (see Chapter 16). All docking tools combine a docking engine with a fast scoring function and the recent literature is full of benchmarks addressing the accuracy of one or few docking/scoring scenarios. The three following issues are usually investigated: (i) the capability of a docking algorithm to reproduce the X-ray pose of selected small-molecular-weight ligands [93, 105, 159], (ii) the propensity of fast scoring functions to recognize near-native poses among a set of decoys [56, 203] and to predict absolute binding free energies [56], and (iii) the discrimination of known binders from randomly chosen molecules in VS experiments [36, 93, 159]. However, analyzing all these data for a comparative analysis of available docking tools is very difficult. First, many tools are not easily available. Second, independent studies assessing the relative performance of docking algorithms/scoring functions are still rare and focus on the use of few methods. Third, the quality judgment may vary depending on the examined properties (quality of the top-ranked pose, quality of all plausible poses, binding free energy prediction, virtual screening utility). Fourth, most docking programs assume approximation levels that can vary considerably [74] and lead, for example, to very inhomogeneous docking paces ranging from few seconds to few hours. Last, many docking programs have been calibrated and validated on small protein–ligand data sets. Hence, detailed benchmarks (above 100 Protein Data Bank–ligand complexes) are only reported for few docking tools [28, 43, 108, 149, 151, 196]. The most recent validation studies on different data sets agree to conclude that the accuracy of a docking tool is largely target dependent [74, 96, 159, 203] and should be examined on a case-by-case basis. Glide and Gold seem to be the most robust programs for their propensity to generate near-native poses in around 75–80% [96, 159], provided that several solutions are stored. A major problem is that the scoring function does not always predict the correct solution as the most probable one (only in around 40–50 % of the cases). This considerably complicates the analysis of docking results. Numerous reasons explain this limited accuracy [93]. Some are easy to correct (e.g. incorrect atom type for either the ligand or the protein), some are more difficult (e.g. accuracy of the protein 3-D structure, flexibility of the ligand, accuracy of the scoring function) and some are really tricky to overcome (protein flexibility, role of bound water). The accuracy of a docking program to predict the protein-bound ligand pose is reflected in its VS efficacy, i.e. the ability to discriminate true binders from inactives and/or randomly chosen compounds [36,93,159]. However, predicting which docking program will be the most suited for a research project is still problematic. If known ligands are available, a pragmatic approach is to try a systematic combination of docking/scoring parameters and select for productive screening the one that best segregates true actives from true inactives. If no or very few ligands are available, some guides may be followed to choose the tool that seems the most appropriate regarding the physicochemical properties of the protein cavity [93].

5.1.5 Which Scoring Function?

The scoring function still remains the Achilles' heel of structure-based VS. Several recent and independent studies conclude that many fast scoring functions can indeed distinguish near-native poses (RMSD < 2.0 Å from the X-ray pose) from decoys for around 70% of high-resolution protein–ligand X-ray structures [56, 203]. However, when docking is applied to a large database, the corresponding scoring function should be robust enough to rank putative hits by increasing binding free energy values [97]. Unfortunately, an accurate prediction of absolute binding free energies is still impossible whatever the method [36, 56, 204]. Predicting binding free energy changes is possible at the condition that a customized scoring function is applied to a series of congeneric ligands. However, for a database containing a large diversity of compounds, and for targets which have not been traditionally used for calibrating scoring functions, the obtained accuracy is usually limited (around 7 kJ mol^{-1} or 1.5 pK units) [75]. From this observation, two sources of improvement are possible: (i) design more accurate scoring functions [204] or (ii) design smarter strategies to postprocess docking outputs (see next section). Many computational chemists actually favor the second option. The accuracy of scoring functions has leveled off several years ago, for the simple reason that some unknown parameters (e.g. role of bound water, protein flexibility) remain extremely difficult to predict whatever the physical principles used to derive a scoring function.

5.1.6 Which Postprocessing?

Acknowledging that scoring functions are far from being perfect, the easiest way to retrieve true positives from a VS is to first detect false positives. Many strategies are possible. The simplest consists in rescoring poses with additional scoring functions; hoping that a consensus scoring [15, 28] will better identify true hits (top-ranked by several scoring functions) from decoys (see Section 4.3). Comparing hit rates between simple and consensus scoring should, however, be realized on hit lists of comparative size [217]. Moreover, customizing a consensus scoring scheme requires first the knowledge of several and chemically diverse true hits. Such data are not always available. Therefore, for less well-investigated targets, other strategies have to be designed. Topological filters can be used to filter out poses exhibiting steric or electrostatic mismatches between the ligand and its target [184]. Poses can also be minimized by a more accurate force field [90, 186], hierarchically clustered [154] or analyzed by Bayesian statistics [103]. In any case, the postprocessing treatment should be simple enough to be reproducible for a wide array of targets. The influence of different postprocessing strategies on the hit rate and the percentage of true hits recovered is shown in Figure 4 (the top-right corner with a hit rate of 100% and all true hits recovered would be the optimum).

An alternative strategy for postprocessing is to look at enrichment among true hits in pre-computed substructures/scaffolds [147]. This presents the advantage of focusing more on scaffolds and the distribution of docking

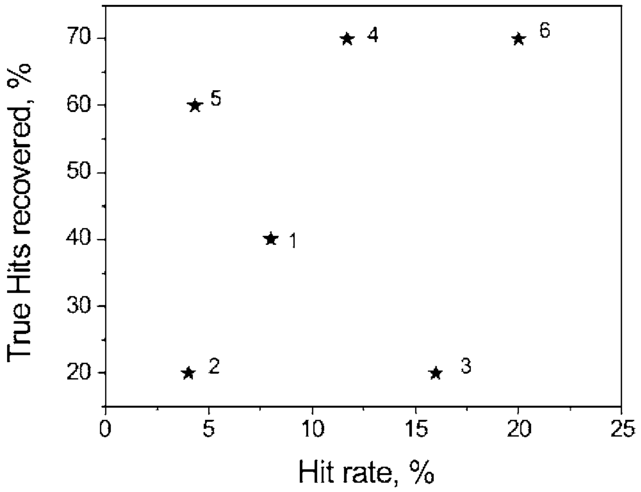


Figure 4 Influence of postprocessing strategies in retrieving true vasopressin V1a receptor antagonists by structure-based screening of a database of 990 randomly chosen "drug-like" compounds seeded with 10 true actives [16]. (1) Top 5% ligands as scored by FlexX; (2) top 5% ligands as scored by Gold; (3) hits common to (1) and (2); (4) ClassPharmer [Bioreason (http://www.bioreason.com)] prioritization of scaffolds for which 60% of the representatives have a FlexXscore lower than $-22\,\text{kJ}\,\text{mol}^{-1}$; (5) ClassPharmer prioritization of scaffolds for which 60% of the representatives have a Goldscore higher than 37.5; (6) ClassPharmer prioritization of scaffolds for which 60% of the representatives have a FlexX score lower than $-22\,\text{kJ}\,\text{mol}^{-1}$ and a Goldscore higher than 37.5.

scores among them, and less on individual molecules. The effect is evident from Figure 4, where the results of such a postprocessing are closer to the optimal corner. Therefore, false negatives may be recovered if they share a scaffold with true positives. Last, but not least, selected hits should be browsed in 3-D target space for the ultimate selection: no algorithms yet outperform the brain of an experienced modeler for such a task!

5.2 Recent Success Stories

Only recent reports from the literature (2003–2005) will be reviewed herein. Most of them still make use of high-resolution X-ray structures. (Sections 5.2.1 –5.2.3). However, encouraging data begin to emerge from homology models (Section 5.2.5) and thus broaden the application of structure-based screening methods to a wider array of pharmaceutically-interesting targets.

5.2.1 **Some Privileged Targets**

Macromolecular targets presenting a well-defined hydrophilic pocket for which the directionality of intermolecular interactions play a key role in ligand recognition are particularly well suited for VS for the simple reason that most docking tools and scoring functions have been calibrated for such situations [56]. Thus, it is no surprise that some protein families (e.g. kinases) are overrepresented in targets for which true inhibitors have been discovered by database docking (Table 1).

Protein kinases have been heavily investigated by structure-based VS [59, 83, 129, 158, 190, 191] to identify novel inhibitors for three major reasons: (i) kinases are among the most relevant target families for the pharmaceutical industry, (ii) a wide array of high-resolution protein–ligand X-ray structures is available for validation purposes and (iii) a canonical hydrogen-bonding to the so-called "hinge region" of the kinase is a typical hallmark of ATP-competitive inhibitors. Two recent studies [129, 191] are representative of the results which might be expected for kinase inhibitors. Vangrevelinghe and coworkers reported a knowledge-based VS protocol for identifying casein kinase II (CK2) inhibitors, in which post-docking filters were designed to downsize the hitlist [191]. Starting from around 400 000 compounds which were docked using Dock4.01 to the 3-D structure of human CK2, 12 000 molecules were first retrieved by score. This primary hitlist was then reduced to 1,592 molecules by selecting only hits which were hydrogen-bonded to the "hinge segment" of the protein and well scored by a consensus scoring function. Visual check of the remaining hits afforded a hit list of only 12 compounds out of which three molecules inhibited the enzyme with an IC_{50} lower than 10 μM.

Pre-docking filters may be useful as well in selecting the most interesting compounds by similarity to known chemotypes present in kinase inhibitors. A good illustration of this strategy has recently been reported by Lyne and coworkers in the discovery of checkpoint kinase-1 inhibitors [129]. A hierarchical screening protocol involving filters of increasing complexity (simple molecular descriptors, 3-D pharmacophore search, FlexX-Pharm constrained docking, knowledge-based consensus scoring) decreases the number of virtual hits from 400 000 to 103, and allowed to identify novel inhibitors in four chemical series. Interestingly, most true inhibitors were not recovered among the top-ranked poses, but by rescoring at least the top 50 poses by a consensus scoring protocol designed from a surrogate kinase (Cdk-2) and a test data set. Post-docking filtering by similarity to well-defined intermolecular interactions may also be a reliable option as it was recently shown to outperform consensus scoring in identifying protein kinase B inhibitors [59]. In the above-cited cases, a precise knowledge-based selection of the most reliable compounds has been achieved thanks to the large information available for related compounds.

The same remark applies to three recent studies aimed at discovering inhibitors of two reductases (dihydrofolate reductase, aldose reductase) [106,

Table 1 Successful structure-based screening data from the recent literature (2003–2005).

Target	Docking	Library	Size	Hit rate[a]	Reference
Chk-1 kinase	FlexX	AstraZeneca	550000	36% @ 68μM	[197]
Casein kinase II	Dock	Novartis	450000	33% @ 10μM	[191]
BCR–ABL	Dock	Chemdiv	200000	13% @ 30μM	[158]
$p56^{Lck}$	Dock	NA[b]	2000000	17% @100μM	[83]
EphB 2	Gold	Chemdiv	50452	5% @ 10μM	[190]
Protein kinase B	FlexX	Chembridge	50000	10% @ 20μM	[59]
DHFR (*Staphylococcus aureus*)	FlexX	Roche	9448	21% @ 25μM	[215]
DHFR	Dock	ACD[c]	NA	33% @ 20μM	[167]
Aldose reductase	FlexX	ACD	260000	55% @ 20μM	[106]
XIAP	Dock	TCM[d]	8000	3% @ 5μM	[148]
Stat3β	Dock	four collections	429000	1% @ 20μM	[183]
Ribosomal A-site	RiboDock	Vernalis collection	1000000	26% @500μM	[58]
IMPDH	FlexX	Roche	3425	8% @100μM	[162]
L-xylose reductase	Dock	NCI[e]	249071	5% @100μM	[26]
PDE4D	FlexX	combinatorial library	320	55% @100nM	[109]
Thymidine phosphorylase	Dock	NCI	250000	7% @ 20μM	[138]
t-RNA guanine-transglycosylase	FlexX	seven collections	827000	55% @ 10μM	[22]
P450 2D6	Gold	NCI subset	111	39% @ 10μM	[95]
SHBG	Glide	natural compounds	23836	7% @ 25μM	[29]
TMPKmt	FlexX	CMC[f] + KEGG [g]	7986	10% @ 20μM	[45]
AICAR-transformylase	AutoDock	NCI	1990	51% @ 20μM	[127]
5-HT$_{1A}$ receptor	Dock	> 20 suppliers	1600000	21% @ 5μM	[13]
NK$_1$ receptor	Dock	> 20 suppliers	1600000	15% @ 5μM	[13]
D$_2$ receptor	Dock	> 20 suppliers	1600000	17% @ 5μM	[13]
CCR$_3$ receptor	Dock	> 20 suppliers	1600000	12% @ 5μM	[13]
5-HT$_4$ receptor	Dock	> 20 suppliers	1600000	21% @ 5μM	[13]
α_{1a} receptor	Gold	Aventis	NA	30% @ 1μM	[54]
NK$_1$ receptor	FlexX	seven collections	827000	14% @ 1μM	[53]
D$_3$ receptor	LigandFit	NCI	250000	40% @ 1μM	[192]

[a] Hit rate at a concentration threshold. The hit rate is the ratio of the number of active compounds to the total number of compounds tested. [b] Not available. [c] Available Chemicals Directory (http://www.mdli.com/products/experiment/available_chem_dir).
[d] Traditional Chinese Medicine Database (http://www.tcm3d.com). [e] National Cancer Institute (http://129.43.27.140/ncidb2). [f] Comprehensive Medicinal Chemistry Database (http://www.mdli.com/products/knowledge/medicinal_chem). [g] KEGG database (http://www.genome.jp/kegg/ligand.html).

167, 215], extensively studied in the past. Wyss and coworkers [215] docked a library of 2,4-diaminopyrimidines to the X-ray structure of DHFR from

S. aureus complexed with an in-house inhibitor. In total, 252 out of the 300 top-ranked compounds could be synthesized and tested for DHFR inhibition; 21 % of the proposed compounds inhibited DHFR from either *S. aureus* or *S. pneumoniae* with IC_{50} values lower 10 µM. Remarkably, a structure-based screening protocol was found to be much superior to a ligand-based diversity selection in enriching a hit list in true inhibitors.

Rastelli and coworkers [167] screened a subset of the ACD for inhibitors of the DHFR from *Plasmodium falciparum* which would be insensitive to specific active site mutations. The full data set was first filtered by Catalyst [Accelrys Software (http://www.accelrys.com/products/catalyst)] to retrieve molecules satisfying a set of 3-D pharmacophores generated from known protein-inhibitor X-ray structures and potentially able to bind to some enzyme mutants. Docking the focussed data set using Dock, then selecting the top-ranked molecules interacting with a key residue and clustering by chemotypes afforded a final list of 24 molecules. Twelve compounds truly inhibited DHFR wild-type as well as active-site mutants at micromolar concentrations.

Krämer and coworkers [106] identified, from the ACD, aldose reductase inhibitors by a series of hierarchical filters implying substructure similarity search to known inhibitors, 2-D pharmacophore filtering, FlexX docking and DrugScore scoring. Compounds able to bind to the anionic pocket of the enzyme were prioritized for purchase and experimental evaluation. Out of the nine compounds tested, six exhibited micromolar inhibition of the target. Interestingly, DrugScore values were weighted according to the molecular weight and number of rotatable bonds of the corresponding molecules to favor the selection of lead-like compounds.

5.2.2 **First-in-class Compounds**

Not all targets are suited for experimental HTS. However, if 3-D coordinates are available, VS is still a cheap alternative to HTS. Two recent studies [148, 183] demonstrate the power of VS for quasi-orphan targets (XIAP, Stat3) of interest for discovering new antitumoral drugs. The X-ray structure of XIAP complexed to a peptidic inhibitor was used to identify, within a database of 8000 compounds derived from traditional Chinese medicinal herbs, a non-peptidic micromolar XIAP inhibitor [148]. Likewise, 429 000 compounds from various screening collections were docked to the X-ray structure of Stat3 – a signal transducer and activator of transcription. Rescoring the top 10% scored compounds from each data set with X-score [202] yielded 200 compounds, out of which 100 could be purchased and tested for Stat3 inhibition [183]. As in the previous study, obtained hit rates at micromolar concentrations were rather low (a single hit out of 100 compounds tested), but a totally novel compound could be discovered and used as a basis for further improvement.

Nucleic acids have not been widely investigated in structure-based screening approaches mainly because of the lack of accurate scoring functions. Foloppe and coworkers [58] recently reported the successful discovery of bacterial ribosomal A-site ligands by using a docking tool (RiboDock) specifically designed for that purpose [142]. An electronic catalogue of 1 million commercially available compounds was first filtered to select lead-like compounds and further docked to the crystal structure of the *Escherichia coli* ribosomal A-site. Visual inspection of the top 2000 best-scoring compounds yielded a list of 129 molecules which were evaluated by a FRET binding assay. Five compounds, unrelated to the aminoglycoside series, exhibited an apparent inhibition constant lower than 50 µM. This study is promising by widening the scope of application of high-throughput docking to nonprotein targets and more successful applications are expected in the near future thanks to a better parameterization of common docking tools for predicting ligand binding to nucleic acids [42].

5.2.3 Fragment Screening

Fragment screening by X-ray or nuclear magnetic resonance (NMR) [197] is becoming an increasingly popular method for identifying low-molecular-weight leads, which usually show a greater optimization potential than drug-like compounds [80]. Due to the difficulty in correctly ranking docking poses of small fragments, computational screening of low-molecular weight compounds is still in its infancy. Two recent reports [26, 162], however, indicate that this approach might be promising.

Pickett and coworkers reported the discovery of low-molecular inhibitors of inosine 5′-monophosphate dehydrogenase (IMPDH) by virtual needle screening [162]. A test set of 21 true IMPDH inhibitors and two in-house X-ray structures was first used to select the most adequate docking/scoring combination (FlexX docking/ScreenScore scoring). A corporate database of 3425 low-molecular-weight reagents was then docked to both X-ray structures to retrieve, among top-ranked compounds, 100 virtual hits satisfying a visual check. Out of the 74 compounds evaluated for IMPDH inhibition, three molecules exhibited an IC_{50} lower than 35 µM.

Carbone and coworkers [26], although not explicitly looking for fragments, also discover low-molecular-weight inhibitors of L-xylose reductase by structure-based screening. Hence, this enzyme is characterized by a very shallow active site and most known xylose reductase (XR) inhibitors are short-chain fatty acids. By screening with the Dock program around 240 000 compounds from the NCI data set [National Cancer Institute, Enhanced NCI Database browser (http://129.43.27.140/ncidb2)] against the X-ray structure of XR, a limited number of putative hits (around 1000) could be prioritized by score and known interactions to key catalytic residues. Out of 39 molecules that were purchased and evaluated for XR inhibition, two carboxylic acids (nicotinic acid, benzoic acid) inhibited the target with IC_{50} values under 100 μM.

Chapter 16 discusses methodical aspects of fragment-based drug design.

5.2.4 Lead Optimization

A large majority of structure-based screening projects are aimed at identifying hits. However, lead optimization might be possible under the condition that the binding mode of the starting lead can be unambiguously recovered, and that a rationale exists for selecting the next compounds to synthesize and test. Krier and coworkers [109] recently proposed a straightforward approach for exploring a lead series by enumerating small-sized libraries (a few hundred compounds) in which all combinatorial assemblies of a few linkers and pharmacophoric moieties to a given scaffold are probed. The selection of the best analogs was based on FlexX docking to the X-ray structure of the phosphodiesterase target and topological filtering. A single-round screening campaign on nine synthesized analogs yielded to a subnanomolar inhibitor and a 900-fold improvement in affinity over the starting lead. Lead optimization is discussed in detail in Chapter 19.

5.2.5 Homology Models as VS Targets

All the above-reported applications have used high-resolution X-ray structures to represent the 3-D coordinates of the target under study. However, enzymes for which a crystal or an NMR structure are still missing, but which show enough sequence homology (around 50%) in the active site to a X-ray template, can also be used for database docking approaches with reasonable success [138]. However, there is still a debate whether targets ranging in a much lower homology range (below 30%) might be reliable starting points. This observation is particularly relevant for GPCRs, a target family of utmost pharmaceutical interest for which a single X-ray structure (bovine rhodopsin) might be used for comparative modelling. Several recent reports [13,16,53,54, 192] demonstrated that GPCRs might be suitable indeed for structure-based screening. In all the above-cited successful cases, preliminary knowledge about known ligands was necessary to fine tune the receptor model. Moreover, the choice of a relevant pharmacophore hypothesis was a key factor to downsize the number of molecules for docking. Last, a visual inspection was necessary to ensure that key intermolecular interactions were established with selected hits. Although the derived homology models remain crude with respect to high-resolution X-ray structures, drug-like submicromolar antagonists for rhodopsin-like receptors [13,16,53,54,192] have already been discovered by VS.

5.3 Concluding Remarks

VS of compound libraries by high-throughput docking is nowadays a routinely used computational technique for identifying bioactive ligands with numerous proofs of record. One should, however, keep in mind that the method is highly sensitive to the 3-D coordinates of the target and is likely to generate numerous false positives. As important as the docking itself are the pre- and postprocessing steps, which are key factors to optimize the hit rate. The number of new validated chemotypes amenable to optimization is therefore a better descriptor than the simple hit rate, which varies considerably with regard to the current knowledge on a particular target. VS is a natural complement to traditional medicinal chemistry and particularly well suited for proposing new molecular scaffolds that can be easily converted into focussed ligand libraries of higher values. Both methodological improvements (scoring, hit triage, prediction of ADMET properties) and better screening collections (focussed and targeted libraries) should contribute to improve the value of this powerful tool in a near future.

6 Critical Evaluation of Ligand-based VS

The choice of tools for ligand-based VS is at least as complicated as for structure-based techniques. While for structure-based techniques mainly the "right" docking program has to be chosen, for ligand-based VS also the method itself, e.g. similarity search or pharmacophore search, has to be chosen. The first part of this section will evaluate and compare several methods and give guidance for selecting appropriate methods and/or tools for the screening. The second part will then, as in Section 5, review some recent success stories and, finally, will draw some conclusions on which methods to apply.

6.1 Influence of Parameter Settings

For ligand-based VS the considerations about the choice of the library and its preprocessing are identical to those for structure-based screening (see Section 5). The main selection process is then, which method among those of Section 3 is chosen. This depends mainly on the number of known active ligands available. If at least some (more than five, better more than 20) ligands are known, a ligand-based pharmacophore model can be derived. If additionally their activity is known, QSAR techniques are possible. If, however, less actives are known, only similarity searches can be performed at this stage.

6.2 Recent Success Stories

Here, we review recent studies from the literature (2003–2005) covering the entire field of ligand-based techniques (Table 2), ranging from pharmacophore searching over similarity searching to QSAR studies.

Table 2 Successful ligand-based screening data from the recent literature (2003–2005).

Target	Method	Library	Size	Hit rate[a]	Reference
ERG2	pharmacophore	WDI[b]	48405	55% @ 1 μM	[117]
σ_1 receptor	pharmacophore	WDI	48405	63% @ 1 μM	[117]
Emopamil binding protein	pharmacophore	WDI	48405	73% @ 1 μM	[117]
Kv1.5	pharmacophore	Aventis	NA[c]	6% @ 6 μM	[161]
A_{2A} purinergic receptor	pharmacophore similarity	combinatorial library	192	53% @ 10nM	[179]
mGluR5	pharmacophore similarity	Asinex	194563	33% @ 70μM	[168]
Tat-TAR RNA interaction	pharmacophore similarity	SPECS library	229659	11% @500μM	[169]
H_1 receptor	QSAR	combinatorial library	9000	87% Watanabe[d]	[48]
Trichomonas vaginalis	QSAR	in-house	100	25% *in vitro*[e]	[140]
Kv1.5	similarity	Aventis	NA	1% @ 10μM	[160]
MCH-1R	similarity[f]	24 collections	650000	2% @ 30μM	[30]
D_3 receptor	pharmacophore fingerprints	two collections	255286	40% @100nM	[25]
COX-2	pharmacophore fingerprints	commercial collections	2700000	15% @ 10μM	[60]

[a] Hit rate at a concentration threshold. The hit rate is the ratio of the number of active compounds to the total number of compounds tested.
[b] World Drug Index (http://scientific.thomson.com/products/wdi).
[c] not available
[d] Antihistaminic activity according to the protocol of Watanabe and coworkers [207].
[e] Cytocidal activity of 100% after 48 h at a concentration of 100 μg ml^{-1}.
[f] A combination of 2- and 3-D substructure search 2- and 3-D similarity as well as clustering was used.

The studies of Laggner and coworkers [117] and Peukert and coworkers [161] demonstrate the application of ligand-based pharmacophore models in VS. Laggner and coworkers [117] built pharmacophore models for ERG2, the emopamil binding protein (EBP) and the σ_1 receptor using Catalyst. The training set comprised 23 structurally diverse ligands with a broad activity range from picomolar to micromolar affinity. The pharmacophore models were assessed using cost analysis and randomization tests. Furthermore, on a test set of nine molecules with binding affinities from subnanomolar to micromolar, from 26 measured affinities, 14 were predicted within 1 order of magnitude. The pharmacophore models were then used for VS of the WDI. From the WDI, previously known binders were found as expected, but also a number of new hits. Among them, 11 were experimentally tested and hit rates between 55 and 75% were obtained for the three targets. Subsequently, the pharmacophore models were altered to perform a search in a subset of the KEGG database of 3525 metabolites. Peukert and coworkers [161] described the discovery of novel blockers of the Kv1.5 potassium ion channel based on pharmacophore search. The authors used DISCO for pharmacophore elucidation using a training set of seven known Kv1.5 blockers. The pharmacophore model obtained was consistent with published SAR data and was able to retrieve 58% of a testset of 423 known Kv1.5 blockers. A 3-D search was performed on the Aventis compound collection resulting in 1975 hits after filtering. In a subsequent clustering, 27 clusters were obtained and representatives of 18 clusters were screened *in vitro*. One active compound was found with an IC_{50} of 5.6 μM belonging to a new class exhibiting a favorable pharmacokinetic profile.

Schneider and Nettekoven [179] demonstrated the use of a topological pharmacophore similarity model named CATS [178]. This approach was applied to the prediction of selective purinergic receptor (A_{2A}) antagonists from a virtual combinatorial library. From a preliminary SAR model an ANN [self-organizing map (SOM)] was trained. Molecules were encoded by the CATS descriptor and the features were mapped from 150-D space onto the plane of a SOM. Each field of the SOM has thus certain pharmacophore features in common. With this technique, the library was reduced from 192 to 17 combinatorial products. These 17 molecules exhibit 3-fold higher binding affinities and 3.5-fold higher selectivities than the initial library. The most selective antagonist displays 121-fold selectivity and an affinity of 2.4 nM. The CATS3D descriptor, a 3-D extension of the CATS approach, was used by Renner and coworkers [168] to identify metabotropic glutamate receptor 5 (mGluR5) modulators. From the original library of 194 563 molecules, first, the 20 000 most "drug-like" compounds were selected and screened by similarity of the CATS3D vectors with each of seven active molecules. Of the obtained 27 top-scoring molecules, nine exhibited an activity below 70 μM. The authors validate that the method used allows for pharmacophore-based similarity searching with "scaffold-hopping". This descriptor was also reported to be successful for identification of new inhibitors of the Tat–TAR RNA interaction [169]. In addition, a "fuzzy" pharmacophore approach (SQUID) was also used. Again, the 20 000 most "drug-like" compounds of an initial library of 229 658 compounds were screened. In the VS the similarities were calculated by the Manhattan distance for the CATS3D and a similarity score for the SQUID, respectively. Both techniques revealed 10 hits, with one molecule overlap. Two molecules among them had IC_{50} values of 500 and 46 μM, respectively.

A screening for antihistaminic compounds blocking the H_1 receptor was performed by Duart and coworkers [48] using a QSAR model based on molecular topology descriptors. From the initial virtual library of 9000 compounds, 236 molecules were predicted as active. Of the selected seven most promising compounds, experimental testing exhibited antihistaminic activity in 87%. The discovery of trichomonacidal compounds was reported by Meneses-Marcel and coworkers [140]. A linear discrimination analysis (LDA) QSAR model was trained to classify molecules using atom-based quadratic indices as descriptors. Since validation of the model revealed 88% good classification, a virtual screening was performed. Biological assays of eight compounds selected by screening gave good classification. Two molecules maintained their efficacy against *T. vaginalis* even at 10 μg ml^{-1} and one of them did not show cytotoxic effects in macrophage cultivations.

A 2-D similarity search with Unity was performed by Peukert and coworkers [160] for blockers of the Kv1.5 ion channel. Using a compound with an IC_{50} of 0.1 μM as reference molecule, 75 compounds with a similarity value of 0.8 or greater were found in the Aventis compound library and experimentally tested. In this step a moderately active compound (IC_{50} = 9.5 μM) was discovered. Although this compound was rejected due to problems with its stability and properties, a compound with similar side-chains, but a different scaffold (naphthene spacer replaced by a biphenyl group), was identified as lead (IC_{50} = 4.8 μM).

Clark and coworkers performed substructure and similarity searches, both in 2- and 3-D, among other techniques, for discovering MCH-1R antagonists. As query compounds 11 known MCH-1R antagonists were selected. The combined hits from all searches were selected (3015 molecules) and assessed for drug-likeness, synthetic tractability and molecular properties. After duplicate removal, 1490 compounds remained which were clustered using Daylight fingerprints. After final visual inspection, 795 compounds were purchased and biochemically screened, resulting in 19 compounds with IC_{50} values below 1 μM and the best having an IC_{50} of 50 nM. Clark and coworkers analyzed, which of the searches revealed which of the 19 compounds. Six compounds were found by 3-D similarity search with FlexS only, also six were found by 3-D substructure search only, two were found by a clustering approach only and one was revealed only by 2-D similarity search. Just four compounds were discovered by more than a single technique. The hit rates were in the range of 0.0 (2-D substructure) to 5.6% (2-D similarity).

6.3 Comparison of Structure- and Ligand-based Techniques

Recently groups at Roche [14], Aventis [52] and Argenta Discovery [30] compared structure- and ligand-based techniques for VS for GPCR targets. Bissantz and coworkers [14] performed a comparative evaluation of the techniques for searching 5-HT$_{2C}$ agonists, while Evers and coworkers [52] performed the comparison on four different biogenic amine-binding GPCRs (α1A, 5-HT$_{2A}$, D2 and M1 receptors), and Clark and coworkers [30] used a number of ligand-based techniques (see above) and compared them to structure-directed pharmacophores.

In the work of Bissantz and coworkers, the results of docking into homology models with FRED were compared with results from Daylight fingerprints, feature trees and the program Phacir. The performance was assessed by hit rate, enrichment factor and the diversity of the structure retrieved. The test database was a collection of actives and inactives from the Roche compound depository, with high similarity between actives and inactives. Four molecules were used as reference for the three similarity search programs (so in total 12 similarity searches were performed) and the top 20% of the ranking lists were analyzed. Each of the 207 actives was retrieved with at least one of the methods by combining the results for each of the reference molecules. When looking at the 12 screening runs, in each individual search many compounds were not retrieved or, even worse, not a single compound with some of the scaffolds was found. Furthermore, the results show that the success of the methods depends strongly on the choice of the reference ligand. While all three similarity measures obtained hit rates of at least 4.8% and enrichment factors of 2.3 or greater for one of the ligands, for another reference ligand the best hit rate was only 2.8%. Some combinations of method and reference ligand did not perform better than random selection. For comparison, the docking program FRED was applied using different scoring functions. The hit rate was between 3.0 and 4.5% and the enrichment factor between 1.5 and 2.2. Thus, while the top-performing ligand-based techniques reached better hit rates than docking, docking always performed better than half of the ligand-based screening runs. Furthermore, the compounds retrieved by structure-based techniques were more diverse on average than those from ligand-based screening. The authors conclude that the results of structure-based screening are more stable than those of ligand-based screening. The latter can yield higher hit rates, but only for some of the reference ligands. Based on these results, the authors propose to combine at least one similarity search with a docking technique.

Evers and coworkers [52] also compared docking into homology models to ligand-based protocols. For the latter, ligand-based pharmacophores, multiple feature trees (MTrees) as well as 3-D similarity by FlexS and QSAR models were applied. Pharmacophore and MTree models were compared on two different reference ligands (one for each class of ligand molecules) for each GPCR. In this study, ligand-based pharmacophore, MTrees and 2-D-QSAR techniques received higher enrichment factors than docking into the homology model with GOLD and FlexX-Pharm. However, the results with GOLD were still satisfying. The authors conclude that docking into GPCR homology models can be useful if no or only a few active ligands are known. In this study the hit rates obtained with FlexS are worse than those obtained from the other virtual screening techniques applied. This is in contrast to results of other studies (e.g. Ref. [30], see above and below) where FlexS gives respectable results. Evers and coworkers conclude that a "fair" comparison can be made only by using several reference structures for the queries.

Clark and coworkers [30] compared a set of different ligand-based methods (see above) to searches using structure-directed pharmacophores. The pharmacophores were generated by docking one ligand into a homology model, then aligning nine other molecules with GASP on the docked conformation and refining the complexes by simulated annealing. Based on this alignment, three different pharmacophore hypotheses were derived and used as queries, but none of them gave rise to a hit.

Ligand- and structure-based VS has not only been compared for GPCR targets. Another group at Aventis used a number of techniques for the search for Kv1.5 ion channel blockers. Apart from the ligand-based work (ligand-based pharmacophore and 2-D similarity, see above) a structure-based screening was also performed in which a protein-derived pharmacophore based on a homology model was used [163]. The structure-based VS gave a higher hit rate (7.8%) than the screenings based on ligand-based pharmacophore (5.5%) and similarity searches (2.7%). Furthermore, the structure-based technique yielded more active compounds and more chemotypes. Even more important is the result that there was no overlap between the hit lists obtained by ligand- and structure-based approaches.

6.4 Concluding Remarks

From the large number of successful applications of ligand-based VS, two general and simple rules can be derived. These rules help to reduce the false-negative rate (ligands being active but not found) of the screening.

(i) *Use as many query ligands as possible.* Several authors have reported that some ligands perform very poorly, not giving any hits at all, while with other reference ligands many hits were found. Unfortunately, it cannot be determined in advance, which of the ligands will be successful.

(ii) *Use as many different techniques as possible.* While some of the hits are "easy" to find by many different techniques, often valuable compounds (e.g. unique scaffolds) are found only by one of the techniques. Again, it cannot be predicted which of the techniques will be successful. It is important to note that not necessarily the most sophisticated techniques yield the most hits. In some cases a very simple search technique can find an interesting compound.

Comparing ligand- and structure-based techniques is difficult since the effectiveness of ligand- and structure-based techniques depends strongly on the screening project. For some targets the ligand-based techniques perform better than structure-based methods, while for other targets they perform worse. From this finding, a third rule can be derived:

(iii) *Use both ligand- and structure-based techniques if possible.* In this combined scenario the maximal benefit of the different starting points can be obtained, and the best compromise between the strengths and limitations of the various methods can be obtained. In other words, make use of the complementarity between ligand- and structure-based techniques [14].

Acknowledgments

The authors are grateful to Matthias Rarey (Hamburg) and Andreas Steffen (Saarbrücken) for valuable suggestions. A large part of this chapter is based on the course "Computer-aided Drug Design" taught by A. K., and he would like to thank the Max Planck Institute for Informatics and the Center for Bioinformatics of Saarland University for the opportunity to give this course.

References

1 Agrafiotis, D. K., V. S. Lobanov, D. N. Rassokhin and S. Izrailev. 2000. The measurement of molecular diversity. In Böhm, H.-J. and G. Schneider (eds.), *Virtual Screening for Bioactive Molecules*. Wiley-VCH, Weinheim: 265–300.

2 Agrafiotis, D. K. and V. S. Lobanov. 2000. Nonlinear mapping networks. J. Chem. Inf. Comput. Sci. **40**: 1356–62.

3 Ahlberg, C. 1999. Visual exploration of HTS databases: bridging the gap between chemistry and biology. Drug Discov. Today **4**: 370–6.

4 Ajay, G., W. Bemis and M. A. Murcko. 1999. Designing Libraries with CNS activity. J. Med. Chem. **42**: 4942–51.

5 Ajay, W., P. Walters and M. A. Murcko. 1998. Can we learn to distinguish between 'drug-like' and 'nondrug-like' molecules? J. Med. Chem. **41**: 3314–24.

6 Allen, F. H. 2002. The Cambridge Structural Database: a quarter of a million crystal structures and rising. Acta Cryst. **B58**: 380–8.

7 Bajorath, J. 2002. Integration of virtual and high-throughput screening. Nat. Rev. Drug Discov. **1**: 882–94.

8 Barnard, J. M., G. M. Down and P. Willett. 2000. Descriptor-based similarity measures for screening chemical databases. In Böhm, H.-J. and Schneider, G. (eds.), *Virtual Screening for Bioactive Molecules*. Wiley-VCH, Weinheim: 59–80.

9 Barnard, J. M. and G. M. Downs. 1997. Chemical fragment generation and clustering software. J. Chem. Inf. Comput. Sci. **37**: 141.

10 Baroni, M., G. Costantini, G. Cruciani, D. Riganelli, R. Valigi and S. Clementi. 1993. Generating optimal linear PLS estimation (GOLPE): An advanced chemometric tool for handling 3D-QSAR problems. Quant. Struct.-Act. Relat. **12**: 9–20.

11 Baurin, N., R. Baker, C. Richardson, et al. 2004. Drug-like annotation and duplicate analysis of a 23-supplier chemical database totalling 2.7 million compounds. J. Chem. Inf. Comput. Sci. **44**: 643–51.

12 Bayada, D. M., H. Hamersma and V. J. van Geerestein. 1999. Molecular diversity and representativity in chemical databases. J. Chem. Inf. Comput. Sci. **39**: 1–10.

13 Becker, O. M., Y. Marantz, S. Shacham, et al. 2004. G protein-coupled receptors: In silico drug discovery in 3D. Proc. Natl. Acad. Sci. USA **101**: 11304–9.

14 Bissantz, C., C. Schalon, W. Guba and M. Stahl. 2005. Focused library design in GPCR projects on the example of 5-HT_{2c} agonists: Comparison of structure-based virtual screening with ligand-based methods. Proteins Struct. Func. Bioinf. **61**: 938–52.

15 Bissantz, C., G. Folkers and D. Rognan. 2000. Protein-based virtual screening of chemical databases. 1. Evaluation of different docking/scoring combinations. J. Med. Chem. **43**: 4759–67.

16 Bissantz, C., P. Bernard, M. Hibert and D. Rognan. 2003. Protein-based virtual screening of chemical databases. ii. Are homology models of G-protein coupled receptors suitable targets? Proteins Struct. Func. Genet. **50**: 5–25.

17 Blaney, F. E., P. Finn, R. W. Phippen and M. Wyatt. 1993. Molecular surface comparison: application to molecular design. J. Mol. Graph. **11**: 98–105.

18 Bohacek, R. S., C. McMartin and W. C. Guida. 1996. The art and practive of structure-based drug design: A molecular modeling perspective. Med. Res. Rev. **1**: 3–50.

19 Boström, J., J. R. Greenwood and J. Gottfries. 2003. Assessing the performance of OMEGA with respect to retrieving bioactive conformations. J. Mol. Graph. Model. **21**: 449–62.

20 Boström, J. 2001. Reproducing the conformations of protein-bound ligands: a critical evaluation of several popular conformational searching tools. J. Comput. Aided Mol. Des. **15**: 1137–52.

21 Bravi, G., E. Gancia, P. Mascagni, M. Pegna, R. Todeschini and A. Zaliani. 1997. MS-WHIM, new 3D theoretical descriptors derived from molecular surface properties: a comparative 3D QSAR study in a series of steroids. J. Comput. Aided Mol. Des. **11**: 79–92.

22 Brenk, R., L. Naerum, U. Grädler, H.-D. Gerber, G. A. Garcia, K. Reuter, M. T. Stubbs and G. Klebe. 2003. Virtual screening fur submicromolar leads of tRNA-guanine transglycosylase based on a new unexpected binding mode detected by crystal structure analysis. J. Med. Chem. **46**: 1133–43.

23 Briem, H. and J. Günther. 2005. Classifying "kinase inhibitor-likeness" by using machine-learning methods. ChemBioChem **6**: 558–66.

24 Brown, R. D. and Y. C. Martin. 1996. Use of structure–activity data to compare structure-based clustering methods and descriptors for use in compound selection. J. Chem. Inf. Comput. Sci. **36**: 572–84.

25 Byvatov, E., B. C. Sasse, H. Stark and G. Schneider. 2005. From virtual to real screening form D_3 dopamine receptor ligands. ChemBioChem **6**: 997–9.

26 Carbone, V., S. Ishikura, A. Hara and O. El-Kabbani. 2005. Structure-based discovery of human L-xylulose reductase inhibitors from database

screening and molecular docking. Bioorg. Med. Chem. **13**: 301–12.

27 Carhart, R. E., D. H. Smith and R. Venkataraghavan. 1985. Atom pairs as molecular features in structure–activity studies: definition and applications. J. Chem. Inf. Comput. Sci. **25**: 64–73.

28 Charifson, P. S., J. J. Corkery, M. A. Murcko and W. P. Walters. 1999. Consensus scoring: a method for obtaining improved hit rates from docking databases of three-dimensional structures into proteins. J. Med. Chem. **42**: 5100–9.

29 Cherkasov, A., Z. Shi, M. Fallahi and G. L. Hammond. 2005. Successful *in silico* discovery of novel nonsteroidal ligands for human sex hormone binding globulin. J. Med. Chem. **48**: 3203–13.

30 Clark, D. E., C. Higgs, S. P. Wren, H. J. Dyke, M. Wong, D. Norman, P. M. Lockey and A. G. Roach. 2004. A virtual screening approach to finding novel and potent antagonists at the melanin-concentrating hormone 1 receptor. J. Med. Chem. **47**: 3962–71.

31 Clark, D. E. 1999. Rapid calculation of polar molecular surface area and its application to the prediction of transport phenomena. 1. Prediction of intestinal absorption. J. Pharm. Sci. **88**: 807–14.

32 Clark, R. D. 1997. OptiSim: an extended dissimilarity selection method for finding diverse representative subsets. J. Chem. Inf. Comput. Sci. **37**: 1181–8.

33 Clement, O. O. and A. T. Mehl. 2000. HipHop: pharmacophores based on multiple common-feature alignments. In Güner, O. F. (ed.), Pharmacophore Perception, Development, and Use in Drug Design. International University Line, La Jolla, USA: 69–84.

34 Cramer, R. D., I., D. E. Patterson and J. D. Bunce. 1988. Comparative molecular field analysis (comfa). 1. Effect of shape on binding of steroids to carrier proteins. J. Am. Chem. Soc. **110**: 5959–67.

35 Cruciani, G., M. Pastor and R. Mannhold. 2002. Suitability of molecular descriptors for database mining. A comparative analysis. J. Med. Chem. **45**: 2685–94.

36 Cummings, M. D., R. L. DesJarlais, A. C. Gibbs, M. Venkatraman and E. P. Jaeger. 2005. Comparison of automated docking programs as virtual screening tools. J. Med. Chem. **48**: 962–76.

37 Cummins, D. J., C. W. Andrews, J. A. Bentley and M. Cory. 1996. Molecular diversity in chemical databases: comparison of medicinal chemistry knowledge bases and databases of commercially available compounds. J. Chem. Inf. Comput. Sci. **36**: 750–63.

38 Davies, K. 1996. Using pharmacophore diversity to select molecules to test from commercial catalogues. In Chaiken, I. M. and K. D. Janda (eds.), *Molecular Diversity and Combinatorial Chemistry: Libraries and Drug Discovery*. American Chemical Society, Washington, DC: 309–16.

39 Dean, P. M. and R. A. Lewis. 1999. *Molecular Diversity in Drug Design.* Kluwer, Dordrecht.

40 Dean, P. M. 1994. *Molecular Similarity in Drug Design.* Chapman & Hall, Glasgow.

41 Denk, Z., C. Chuaqui and J. Singh. 2004. Structural Interaction Fingerprint (SIFt): A novel method for analyzing three-dimensional protein–ligand binding interactions. J. Med. Chem. **47**: 337–44.

42 Detering, C. and G. Varani. 2004. Validation of automated docking programs for docking and database screening against RNA drug targets. J. Med. Chem. **47**: 4188–201.

43 Diller, D. J., J. Merz and M. Kenneth. 2001. High throughput docking for library design and library prioritization. Proteins Struct. Func. Genet. **43**: 113–24.

44 DiMasi, J. A., R. W. Hansen and H. G. Grabowski. 2003. The price of innovation: new estimates of drug development costs. J. Health Econ. **22**: 151–85.

45 Douguet, D., H. Munier-Lehmann, G. Labesse and S. Pochet. 2005. LEA3D: a computer-aided ligand design for structure-based drug design. J. Med. Chem. **48**: 2457–68.

46 DOWNS, G. M. AND P. WILLETT. 1994. Clustering of chemical structure databases for compound selection. In VAN DE WATERBEEMD, H. (ed.), *Advanced Computer-Assisted Techniques in Drug Discovery*. VCH, Weinheim: 111–130.

47 DREWS, J. 2000. Drug discovery today – and tomorrow. Drug Discov. Today **5**: 2–4.

48 DUART, M. J., G. M. ANTÓN-FOS, P. A. ALEMÁN, J. B. GAY-ROIG, M. E. GONZÁLEZ-ROSENDE, J. GÁLVEZ AND R. GARCIA-DOMENECH. 2005. New potential antihistaminic compounds. Virtual combinatorial chemistry, computational screening, real synthesis, and pharmacological evaluation. J. Med. Chem. **48**: 1260–4.

49 ENGELS, M. F. M., T. THIELEMANS, D. VERBINNEN, J. P. TOLLENAERE AND R. VERBEECK. 2000. CerBeruS: a system supporting the sequential screening process. J. Chem. Inf. Comput. Sci. **40**: 241–5.

50 ERTL, P., B. ROHDE AND P. SELZER. 2000. Fast calculation of molecular polar surface area as a sum of fragment-based contributions and its application to the prediction of drug transport properties. J. Med. Chem. **43**: 3714–7.

51 EVENSEN, E., J. E. EKSTEROWICZ, R. V. STANTON, C. OSHIRO, P. D. J. GROOTENHUIS AND E. K. BRADLEY. 2003. Comparing performance of computational tools for combinatorial library design. J. Med. Chem. **46**: 5125–8.

52 EVERS, A., G. HESSLER, H. MATTER AND T. KLABUNDE. 2005. Virtual screening of biogenic amine-binding G-protein coupled receptors: comparative evaluation of protein- and ligand-based virtual screening protocols. J. Med. Chem. **48**: 5448–65.

53 EVERS, A. AND G. KLEBE. 2004. Successful virtual screening for a submicromolar antagonist of the neurokinin-1 receptor based on a ligand-supported homology model. J. Med. Chem. **47**: 5381–92.

54 EVERS, A. AND T. KLABUNDE. 2005. Structure-based drug discovery using GPCR homology modeling: successful virtual screening for antagonists of the alpha1A adrenergic receptor. J. Med. Chem. **48**: 1088–97.

55 FEHER, M. AND J. M. SCHMIDT. 2003. Property distributions: differences between drugs, natural products, and molecules from combinatorial chemstry. J. Chem. Inf. Comput. Sci. **43**: 218–27.

56 FERRARA, P., H. GOHLKE, D. J. PRICE, G. KLEBE AND I. BROOKS, CHARLES L.. 2004. Assessing scoring funcitons for protein–ligand interactions. J. Med. Chem. **47**: 3032–47.

57 FINK, T., H. BRUGGESSER AND J.-L. REYMOND. 2005. Virtual exploration of the small-molecule chemical universe below 160 daltons. Angew. Chem. Int. Ed. **44**: 1504–8.

58 FOLOPPE, N., I.-J. CHEN, B. DAVIS, A. HOLD, D. MORLEY AND R. HOWES. 2004. A structure-based strategy to identify new molecular scaffolds targeting the bacterial ribosomal A-site. Bioorg. Med. Chem. **12**: 935–47.

59 FORINO, M., D. JUNG, J. B. EASTON, P. J. HOUGHTON AND M. PELLECHIA. 2005. Virtual docking approaches to protein kinase B inhibition. J. Med. Chem. **48**: 2278–81.

60 FRANKE, L., E. BYVATOV, O. WERZ, D. STEINHILBER, P. SCHNEIDER AND G. SCHNEIDER. 2005. Extraction and visualization of potential pharmacophore points using support vector machines: application to ligand-based virtual screening for COX-2 inhibitors. J. Med. Chem. **48**: 6997–7004.

61 FRANKE, R. AND A. GRUSKA. 1995. Principal component and factor analysis. In VAN DE WATERBEEMD, H. (ed.), *Chemometric Methods in Molecular Design*. VCH, Weinheim: 113–63.

62 GASTREICH, M., H. BRIEM, C. LEMMEN AND M. RAREY. 2005. Addressing the virtual screening challenge – the Flex* approach. In ALVAREZ, J. AND B. SHOICHET (eds.), *Virtual Screening in Drug Discovery*. Decker/CRC Press, Boca Raton, FL: 25–46.

63 GEDECK, P. AND P. WILLET. 2001. Visual and computational analysis of structure-activity relationships in high-throughput screening data. Curr. Opin. Chem. Biol. **5**: 389–95.

64 Ghose, A. K., V. N. Viswanadhan and J. J. Wendoloski. 1999. A knowledge-based approach in desigining combinatorial or medicinal libraries for drug discovery. 1. A qualitative and quantitative characterization of known drug databases. J. Comb. Chem. **1**: 55–68.

65 Gillet, V. J., P. Willett and J. Bradshaw. 1998. Identification of biological activity profiles using substructural analysis and genetic algorithms. J. Chem. Inf. Comput. Sci. **38**: 165–79.

66 Ginn, C. M. R., P. Willett and J. Bradshaw. 2000. Combination of molecular similarity measures using data fusion. Perspect. Drug Discov. Des. **20**: 1–16.

67 Giordanetto, F., S. Cotesta, C. Catana, J.-Y. Trosset, A. Vulpetti, P. F. W. Stouten and R. T. Kroemer. 2004. Novel scoring functions comprising QXP, SASA, and protein side-chain entropy terms. J. Chem. Inf. Comput. Sci. **44**: 882–93.

68 Good, A. C., E. E. Hodgkin and W. G. Richard. 1992. Utilization of Gaussian functions for the rapid evaluation of molecular similarity. J. Chem. Inf. Comput. Sci. **32**: 188–91.

69 Good, A. C., S. R. Krystek and J. S. Mason. 2000. High-throughput and virtual screening: core lead discovery technologies move towards integration. Drug Discov. Today **12**: S61–9.

70 Good, A. C. and D. L. Cheney. 2003. Analysis and optimization of structure-based virtual screening protocola (1): exploration of ligand conformational sampling techniques. J. Mol. Graph. Model. **22**: 23–30.

71 Goodford, P. 1996. Multivariate characterization of molecules for QSAR analysis. J. Chemometrics **10**: 107–17.

72 Griffith, R., T. T. T. Luu, J. Garner and P. A. Keller. 2005. Combining structure-based drug design and pharmacophores. J. Mol. Graph. Model. **23**: 439–46.

73 Güner, O. F. 2000. *Pharmacophore Perception, Development and Use in Drug Design.* International University Line, La Jolla, CA.

74 Halperin, I., B. Ma, H. Wolfson and R. Nussinov. 2002. Principles of docking: an overview of search algorithms and a guide to scoring functions. Proteins Struc. Func. Genet. **47**: 409–43.

75 Hann, M., B. Hudson, X. Lewell, R. Lifely, L. Miller and N. Ramsden. 1999. Strategic pooling of compounds for high-throughput screening. J. Chem. Inf. Comput. Sci. **39**: 897–902.

76 Hansch, C. and A. Leo. 1979. *Substituent Constants for Correlation Analysis in Chemistry.* Wiley, New York, NY.

77 Hansch, C. and T. Fujita. 1964. ρ-σ-π analysis. A method for the correlation of biological activity and chemical structure. J. Am. Chem. Soc. **86**: 1616–26.

78 Hertzberg, R. P. and A. J. Pope. 2000. High-throughput screening: new technology for the 21st century. Curr. Opin. Chem. Biol. **4**: 445–51.

79 Hofbauer, C., H. Lohninger and A. Aszódi. 2004. SURFCOMP: a novel graph-based approach to molecular surface comparison. J. Chem. Inf. Comput. Sci. **44**: 837–47.

80 Hopkins, A. L., C. R. Groom and A. Alex. 2004. Ligand efficiency: a useful metric for lead selection. Drug Discov. Today **9**: 430–1.

81 Hopkins, A. L. and C. R. Groom. 2002. The druggable genome. Nat. Rev. Drug Discov. **1**: 727–30.

82 Höskuldsson, A. 1988. PLS regression methods. J. Chemometrics **2**: 211–28.

83 Huang, N., A. Nagarsekar, G. Xia, J. Hayashi and A. D. MacKerell, Jr.. 2004. Identification of non-phosphate-containing small molecular weight inhibitors of the tyrosine kinase p56 Lck SH2 domain via in silico screening against the pY + 3 binding site. J. Med. Chem. **47**: 3502–11.

84 Ihlenfeldt, W.-D. and J. Gasteiger. 1994. Hash codes for the identification and classification of molecular structure elements. J. Comput. Chem. **15**: 793–813.

85 Irwin, J. J. and B. K. Shoichet. 2005. ZINC – a free database of commercially available compounds for virtual

screening. J. Chem. Inf. Model. **45**: 177–82.

86 ITAI, A., N. TOMIOKA, M. YAMADA, A. INOUE AND Y. KATO. 1993. Molecular superposition for rational drug design. In KUBINYI, H. (ed.), *3D QSAR in Drug Design: Theory, Methods and Applications*. ESCOM, Leiden: 173–99.

87 JOHNSON, M. A. AND G. M. MAGGIORA. 1990. *Concepts and Applications of Molecular Similarity.* Wiley, New York, NY.

88 JONES, G., P. WILLET AND R. C. GLEN. 1995. A genetic algorithm for flexible molecular overlay and pharmacophoere elucidation. J. Comput. Aided Mol. Des. **9**: 532–49.

89 KAHNBERG, P., M. H. HOWARD, T. LILJEFORS, M. NIELSEN, E. Ø. NIELSEN, O. STERNER AND I. PETTERSSON. 2004. The use of a pharmacophore model for identification of novel ligands for the benzodiazepine binding site of the GABA$_A$ receptor. J. Mol. Graph. Model. **23**: 253–61.

90 KALYANARAMAN, C., K. BERNACKI AND M. P. JACOBSON. 2005. Virtual screening against highly charged active sites: identifying substrates of alpha-beta barrel enzymes. Biochemistry **44**: 2059–71.

91 KATO, Y., A. INOUE, M. YANADA, N. TOMICKA AND A. ITAI. 1992. Automatic superposition of drug molecules based on their common receptor site. J. Comput. Aided Mol. Des. **6**: 475–86.

92 KEARSLEY, S. K. AND G. M. SMITH. 1990. An alternative method for the alignment of molecular structures: maximizing electrostatic and steric overlap. Tetrahedron Comput. Methodol. **3**: 615–33.

93 KELLENBERGER, E., J. RODRIGO, P. MULLER AND D. ROGNAN. 2004. Comparative evaluation of eight docking tools for docking and virtual screening accuracy. Proteins Struct. Func. Bioinf. **57**: 225–42.

94 KELLOGG, G. E., S. F. SEMUS AND D. J. ABRAHAM. 1991. HINT – a new method of empirical hydrophobic field calculation for CoMFA. J. Comput. Aided Mol. Des. **5**: 545–52.

95 KEMP, C. A., J. U. FLANAGAN, A. J. VAN ELDIK, et al., 2004. Validation of Model of Cytochrome P450 2D6: an *in silico* tool for predicting metabolism and inhibition. J. Med. Chem. **47**: 5340–6.

96 KIRCHMAIR, J., C. LAGGNER, G. WOLBER AND T. LANGER. 2005. Comparative analysis of protein-bound ligand conformations with respect to Catalyst's conformational space subsampling algorithm. J. Chem. Inf. Model. **45**: 422–30.

97 KITCHEN, D. B., H. DECORNEZ, J. R. FURR AND J. BAJORATH. 2004. Docking and scoring in virtual screening for drug discovery: methods and applications. Nat. Rev. Drug Discov. **3**: 935–49.

98 KLEBE, G., T. MIETZNER AND F. WEBER. 1994. Different approaches toward an automatic structural alignment of drug molecules: applications to sterol mimics, thrombin and thermolysin inhibitors. J. Comput. Aided Mol. Des. **8**: 751–78.

99 KLEBE, G., U. ABRAHAM AND T. MIETZNER. 1994. Molecular similarity indices in a comparative analysis (CoMSIA) of drug molecules to correlate and predict their biological activity. J. Med. Chem. **37**: 4130–46.

100 KLEBE, G. AND T. MIETZNER. 1994. A fast and efficient method to generate biologically relevant conformations. J. Comput. Aided Mol. Des. **8**: 583–606.

101 KLEBE, G. 2000. *Virtual screening: An Alternative or Complement to High Throughput Screening.* Kluwer, Dordrecht.

102 KLON, A. E., M. GICK, M. THOMA, P. ACKLING AND J. W. DAVIES. 2004. Finding more needles in the haystack: a simple and efficient method for improving high-throughput docking results. J. Med. Chem. **47**: 2743–9.

103 KLON, A. E., M. GLICK AND J. W. DAVIES. 2004. Appliation of machine learning to improve the results of high-throughput docking against HIV-1 protease. J. Chem. Inf. Comput. Sci. **44**: 2216–24.

104 KONTOYIANNI, M., G. S. SOKOL AND L. M. MCCLELLAN. 2005. Evaluation of library ranking efficacy in virtual screening. J. Comput. Chem. **26**: 11–22.

105 KONTOYIANNI, M., L. M. MCCLELLAN AND G. S. SOKOL. 2004. Evaluation of docking performance: comparative data on docking algorithms. J. Med. Chem. **47**: 558–65.

106 KRAEMER, O., I. HAZEMANN, A. D. PODJARNY AND G. KLEBE. 2004. Virtual screening for inhibitors of human aldose reductase. Proteins **55**: 814–23.

107 KRÄMER, A., H. W. HORN AND J. E. RICE. 2003. Fast 3D molecular superposition and similarity search in databases of flexible molecules. J. Comput. Aided Mol. Des. **17**: 13–38.

108 KRAMER, B., M. RAREY AND T. LENGAUER. 1999. Evaluation of the FlexX incremental construction algorithm for protein–ligand docking. Proteins Struct. Func. Genet. **37**: 228–41.

109 KRIER, M., J. X. DE ARAÚJO-JÚNIOR, M. SCHMITT, J. DURANTON, H. JUSTIANO-BASARAN, C. LUGNIER, J.-J. BOURGUIGNON AND D. ROGNAN. 2005. Design of small-sized libraries by combinatorial assembly of linkers and functional groups to a given scaffold: application to the structure-based optimization of a phosphodiesterase 4 inhibitor. J. Med. Chem. **48**: 3816–22.

110 KRUSKAL, J. B. 1964. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypotheses. Psychometrika **29**: 1–27.

111 KUBINYI, H., G. FOLKERS AND Y. C. MARTIN. 1998. *3D QSAR in Drug Design*, Vol. 2. Kluwer/ESCOM, Dordrecht.

112 KUBINYI, H., G. FOLKERS AND Y. C. MARTIN. (eds.). 1998. *3D QSAR in Drug Design*, Vol. 3. Kluwer/ESCOM, Dordrecht.

113 KUBINYI, H. 1993. *3D QSAR in Drug Design: Theory, Methods and Applications.* ESCOM, Leiden.

114 KUBINYI, H. 1998. Similarity and dissimilarity: a medicinal chemist's view. Perspect. Drug Discov. Des. **9–11**: 225–52.

115 KUNTZ, I. D., J. M. BLANEY, S. J. OATLEY, R. LANGRIDGE AND T. E. FERRIN. 1982. A geometric approach to macromolecule–ligand interactions. J. Mol. Biol. **161**: 269–88.

116 LADD, B. 2000. Intuitive data analysis: the next generation. Mod. Drug Discov. **3**: 46–52.

117 LAGGNER, C., C. SCHIEFERER, B. FIECHTNER, G. POLES, R. D. HOFFMAN, H. GLOSSMANN, T. LANGER AND F. F. MOEBIUS. 2005. Discovery of high-affinity ligands of σ_1 receptor, ERG2, and emopamil binding protein by pharmacophore modeling and virtual screening. J. Med. Chem. **48**: 4754–64.

118 LAJINESS, M. S. 1991. Evaluation of the performance of dissimilarity performance methodology. In SILIPO, C. AND A. VITTORIA (eds.), *QSAR: Rational Aproaches to the Design of Bioactive Compounds*. Elsevier, Amsterdam: 201–4.

119 LEACH, A. R. 1991. A survey of methods for searching the conformational space of small and medium-sized molecules. In LIPKOWITZ, K. B. AND D. B. BOYD (eds.), *Reviews in Computational Chemistry*. VCH, Weinheim: 1–55.

120 LEMMEN, C., C. HILLER AND T. LENGAUER. 1998. RigFit: a new approach to superimposing ligand molecules. J. Comput. Aided Mol. Des. **12**: 491–502.

121 LEMMEN, C., M. ZIMMERMANN AND T. LENGAUER. 2000. Multiple molecular superpositioning as an effective tool for virtual database screening. Perspect. Drug Discov. Des. **20**: 43–62.

122 LEMMEN, C., T. LENGAUER AND G. KLEBE. 1998. FlexS: a method for fast flexible ligand superposition. J. Med. Chem. **41**: 4502–20.

123 LEMMEN, C. AND T. LENGAUER. 2000. Computational methods for the structural alignment of molecules. J. Comput. Aided Mol. Des. **14**: 215–32.

124 LESSEL, U. F. AND H. BRIEM. 2000. Flexsim-X: a method for the detection of molecules with similar biological activity. J. Chem. Inf. Comput. Sci. **40**: 246–53.

125 LEWIS, R. A., J. S. MASON AND I. M. MCLAY. 1997. Similarity measures for rational set selection and analysis of combinatorial libraries: the diverse property-derived (DPD) approach. J. Chem. Inf. Comput. Sci. **37**: 599–614.

126 Lewis, R. A., S. D. Pickett and D. E. Clark. 2000. Computer-aided molecular diversity analysis and combinatorial library design. In Lipkowitz, K. B. and D. B. Boyd (eds.), *Reviews in Computational Chemistry*. Wiley-VCH, Weinheim: 1–51.

127 Li, C., L. Xu, D. W. Wolan, I. A. Wilson and A. J. Olson. 2004. Virtual screening of human 5-aminoimidazole-4-carboxamide ribonucleotide transformylase against the NCI diversity set by use of AutoDock to identify novel nonfolate inhibitors. J. Med. Chem. **47**: 6681–90.

128 Lipinski, C. A., F. Lombardo, B. W. Dominy and P. J. Feeney. 1997. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Adv. Drug Deliv. Rev. **23**: 3–25.

129 Lyne, P. D., P. W. Kenny, D. A. Cosgrove, C. Deng, S. Zabludoff, J. J. Wendoloski and S. Ashwell. 2004. Identification of compounds with nanomolar binding affinity for checkpoint kinase-1 using knowledge-based virtual screening. J. Med. Chem. **47**: 1962–8.

130 Manallack, D. T., W. R. Pitt, E. Gancia, J. G. Montana, D. J. Livingstone, M. G. Ford and D. C. Whitley. 2002. Selecting screening candidates for kinase and G protein-coupled ceceptor targets using neural networks. J. Chem. Inf. Comput. Sci. **42**: 1256–62.

131 Martin, Y. C., J. L. Kofron and L. M. Traphagen. 2002. Do structurally similar molecules have similar biological activities? J. Med. Chem. **45**: 4350–8.

132 Martin, Y. C., M. G. Bures, E. A. Danaher, J. DeLazzer, I. Lico and P. A. Pavlic. 1993. A fast new approach to pharmacophore mapping and its application to dopaminergic and benzodiazepine agonists. J. Comput. Aided Mol. Des. **7**: 83–102.

133 Martin, Y. C. 1992. 3D database searching in drug design. J. Med. Chem. **35**: 2145–54.

134 McGovern, S. L., B. T. Helfand, B. Feng and B. K. Shoichet. 2003. A specific mechanism of nonspecific inhibition. J. Med. Chem. **46**: 4265–72.

135 McGovern, S. L., E. Caselli, N. Grigorieff and B. K. Shoichet. 2002. A common mechanism underlying promiscuous inhibitors from virtual and high-throughput screening. J. Med. Chem. **45**: 1712–22.

136 McGregor, M. J. and S. M. Muskal. 1999. Pharmacophore fingerprinting. 1. Application to QSAR and focused library design. J. Chem. Inf. Comput. Sci. **39**: 569–74.

137 McMartin, C. and R. S. Bohacek. 1995. Flexible matching of test ligands to a 3D pharmacophore using a molecular superposition force field: Comparison of predicted and experimental conformations of inhibitors of three enzymes. J. Comput. Aided Mol. Des. **9**: 237–50.

138 McNally, V. A., A. Gbaj, K. T. Douglas, L. J. Stratford, M. Jaffar, S. Freeman and R. A. Bryce. 2003. Identification of a novel class of inhibitor of human and *Escherichia coli* thymidine phosphorylase by *in silico* screening. Bioorg. Med. Chem. Lett. **13**: 3705–9.

139 McQueen, J. 1967. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. on Mathematical Statistics and Probability*, Berkeley, CA: 281–97.

140 Meneses-Marcel, A., Y. Marrero-Ponce, Y. Machado-Tugores, et al. 2005. A linear discrimination analysis based virtual screening of trichomonacidal lead-like compounds: outcomes of in silico studies supported by experimental results. Bioorg. Med. Chem. Lett. **15**: 3838–43.

141 Merkwirth, C., H. Mauser, T. Schulz-Gasch, O. Roche, M. Stahl and T. Lengauer. 2004. Ensemble methods for classification in cheminformatics. J. Chem. Inf. Comput. Sci. **44**: 1971–8.

142 Morley, S. D. and M. Afshar. 2004. Validation of an empirical RNA–ligand scoring function for fast flexible docking

using RiboDock. J. Comput. Aided Mol. Des. **18**: 189–08.

143 MORRIS, J. J. AND P. P. BRUNEAU. 2000. Prediction of physicochemical properties. In BÖHM, H.-J. AND G. SCHNEIDER (eds.), *Virtual Screening for Bioactive Molecules*. Wiley-VCH, Weinheim: 33–58.

144 MÜLLER, K.-R., G. RÄTSCH, S. SONNENBURG, S. MIKA, M. GRIMM AND N. HEINRICH. 2005. Classifying "drug-likeness" with kernel-based learning methods. J. Chem. Inf. Comput. Sci. **45**: 249–53.

145 MURPHY, K. P. AND E. FREIRE. 1992. Thermodynamics of structural stability and cooperative folding behavior in proteins. Adv. Protein Chem. **43**: 313–61.

146 MURTAUGH, F. 1983. A survey of recent advances in hierarchical clustering alrorithms. Computer J. **26**: 354–9.

147 NICOLAOU, C. A., S. Y. TAMURA, B. P. KELLEY, S. I. BASSETT AND R. F. NUTT. 2002. Analysis of large screening data sets via adaptively grown phylogenetic-like trees. J. Chem. Inf. Comput. Sci. **42**: 1069–79.

148 NIKOLOVSKA-COLESKA, Z., L. XU, Z. HU, et al. 2004. Discovery of embelin as a cell-permeable, small-molecular weight inhibitor of XIAP through structure-based computational screening of a traditional herbal medicine three-dimensional structure database. J. Med. Chem. **47**: 2430–40.

149 NISSINK, J. W. M., C. MURRAY, M. HARTSHORN, J. C. VERDONK, MARCEL L.AND COLE AND R. TAYLOR. 2002. A new test set for validating predictions of protein–ligand interaction. Proteins Struct. Func. Genet. **49**: 457–71.

150 OSHIRO, C., E. K. BRADLEY, J. EKSTEROWICZ, et al. 2004. Performance of 3D-database molecular docking studies into homology models. J. Med. Chem. **47**: 764–7.

151 PANG, Y.-P., E. PEROLA, K. XU AND F. G. PRENDERGAST. 2001. EUDOC: a computer program for identification of drug interaction sites in macromolecules and drug leads from chemical databases. J. Comput. Chem. **22**: 1750–71.

152 PATEL, Y., V. J. GILLET, G. BRAVI AND A. R. LEACH. 2002. A comparison of the pharmacophore identification programs: Catalyst, DISCO and GASP. J. Comput. Aided Mol. Des. **16**: 653–81.

153 PATTERSON, D. E., R. D. CRAMER, A. M. FERGUSON, R. D. CLARK AND L. E. WEINBERGER. 1996. Neighborhood behavior: a useful concept for validation of "molecular diversity" descriptors. J. Med. Chem. **39**: 3049–59.

154 PAUL, N. AND D. ROGNAN. 2002. ConsDock: a new program for the consensus analysis of protein–ligand interactions. Proteins Struct. Func. Genet. **47**: 521–533.

155 PEARLMAN, R. S. AND K. M. SMITH. 1998. Novel software tools for chemical diversity. Perspect. Drug Discov. Des. **9–11**: 339–53.

156 PEARLMAN, R. S. AND K. M. SMITH. 1999. Metric validation and the receptor-relevant subspace concept. J. Chem. Inf. Comput. Sci. **39**: 28–35.

157 PEARLMAN, R. S. 1993. 3D molecular structures: generation and use in 3D-searching. In KUBINYI, H. (ed.), *3D QSAR in Drug Design: Theory, Methods and Applications*. ESCOM, Leiden: 21–58.

158 PENG, H., N. HUANG, J. QI, P. XIE, C. XU, J. WANG AND C. YANG. 2003. Identification of novel inhibitors of BCL–ABL tyrosine kinase via virtual screening. Bioorg. Med. Chem. Lett. **13**: 3693–9.

159 PEROLA, E., W. P. WALTERS AND P. S. CHARIFSON. 2004. A detailed comparison of current docking and scoring methods on systems of pharmaceutical relevance. Proteins **56**: 235–49.

160 PEUKERT, S., J. BRENDEL, B. PIRARD, A. BRÜGGEMANN, P. BELOW, H.-W. KLEEMANN, H. HEMMERLE AND W. SCHMIDT. 2003. Identification, synthesis, and activity of novel blockers of the voltage-gated potassium channel Kv1.5. J. Med. Chem. **46**: 486–98.

161 PEUKERT, S., J. BRENDEL, B. PIRARD, C. STRÜBING, H.-W. KLEEMANN, T. BÖHME AND H. HEMMERLE. 2004. Pharmacophore-based search, synthesis, and biological evaluation of anthranilic

amides as novel blockers of the Kv1.5 channel. Bioorg. Med. Chem. Lett. **14**: 2823–7.

162 Pickett, S. D., B. S. Sherborne, T. Wilkinson, et al. 2003. Discovery of novel low molecular weight inhibitors of IMPDH via virtual needle screening. Bioorg. Med. Chem. Lett. **13**: 1691–4.

163 Pirard, B., J. Brendel and S. Peukert. 2005. The discovery of Kv1.5 blockers as a case study for the application of virtual screening approaches. J. Chem. Inf. Model. **45**: 477–85.

164 Pitman, M. C., W. K. Huber, H. Horn, A. Krämer, J. E. Rice and W. C. Swope. 2001. FLASHFLOOD: a 3D field-based similarity search and alignment method for flexible molecules. J. Comput. Aided Mol. Des. **15**: 587–612.

165 Rarey, M., B. Kramer, T. Lengauer and B. Klebe. 1996. A fast flexible docking method using an incremental construction algorithm. J. Mol. Biol. **261**: 470–89.

166 Rarey, M. and J. S. Dixon. 1998. Feature trees: a new molecular similarity measure based on tree matching. J. Comput. Aided Mol. Des. **12**: 471–90.

167 Rastelli, G., S. Pacchioni, W. Sirawaraporn, R. Sirawaraporn, M. D. Parenti and A. M. Ferrari. 2003. Docking and database screening reveal new classes of *Plasmodium falciparum* dihydrofolate reductase inhibitors. J. Med. Chem. **46**: 2834–45.

168 Renner, S., T. Noeske, C. G. Parsons, P. Schneider, T. Weil and G. Schneider. 2005. New allosteric modulators of metabotropic glutamate receptor 5 (mGluR5) found by ligand-based virtual screening. ChemBioChem **6**: 620–5.

169 Renner, S., V. Ludwig, O. Boden, U. Scheffer, M. Göbel and G. Schneider. 2005. New inhibitors of the Tat–TAR RNA interaction found with a “fuzzy” pharmacophore model. ChemBioChem **6**: 1119–25.

170 Rishton, G. M. 1997. Reactive compounds and in vitro false positives in HTS. Drug Discov. Today **2**: 382–4.

171 Roberts, G., G. J. Myatt, W. P. Johnson, K. P. Cross and J. Blower, Paul E.. 2000. LeadScope: software for exploring large sets of screening data. J. Chem. Inf. Comput. Sci. **40**: 1302–14.

172 Roche, O., P. Schneider, J. Zuegge, et al. 2002. Development of a virtual screening method for identification of “frequent hitters” in compound libraries. J. Med. Chem. **45**: 137–42.

173 Rusinko, Andres, I., M. W. Farmen, C. G. Lambert, P. L. Brown and S. S. Young. 1999. Analysis of a large structure/biological activity data set using recursive partitioning. J. Chem. Inf. Comput. Sci. **39**: 1017–26.

174 Russ, A. P. and S. Lampel. 2005. The druggable genome: an update. Drug. Discov. Today **10**: 1607–10.

175 Sadowski, J. and H. Kubinyi. 1998. A scoring scheme for discriminating between drugs and nondrugs. J. Med. Chem. **41**: 3325–9.

176 Sadowski, J. and J. Gasteiger. 1993. From atoms and bonds to three-dimensional atomic coordinates: automatic model builders. Chem. Rev. **93**: 2567–81.

177 Sams-Dodd, F. 2005. Target-based drug discovery: is something wrong? Drug Discov. Today **10**: 139–47.

178 Schneider, G., W. Neidhart, T. Giller and G. Schmid. 1999. “Scaffold-hopping” by topological pharmacophore search: a contribution to virtual screening. Angew. Chem. Int. Ed. Engl. **38**: 2894–6.

179 Schneider, G. and M. Nettekoven. 2003. Ligand-based combinatorial design of selective purinergic receptor A_{2A} antagonists using self-organizing maps. J. Comb. Chem. **5**: 233–7.

180 Sheridan, R. P., R. Nilakantan, J. S. Dixon and R. Venkataraghavan. 1986. The ensemble approach to distance geometry: application to the nicitinic pharmacophore. J. Med. Chem. **29**: 899–906.

181 Sheridan, R. P. and S. K. Kerarley. 2002. Why do we need so many chemical similarity search methods? Drug Discov. Today **7**: 903–11.

182 SIROIS, S., G. HATZAKIS, D. WEI, Q. DU AND C. KUO-CHEM. 2005. Assessment of chemical libraries for their druggability. Comput. Biol. Chem. **29**: 55–67.

183 SONG, H., R. WANG, S. WANG AND J. LIN. 2005. A low-molecular-weight compound discovered through virtual database screening inhibits Stat3 function in breast cancer cells. Proc. Natl Acad. Sci. USA **102**: 4700–5.

184 STAHL, M. AND H.-J. BÖHM. 1998. Development of filter functions for protein–ligand docking. J. Mol. Graph. Model. **16**: 121–32.

185 STANTON, D. T., T. W. MORRIS, S. ROYCHOUDHURY AND C. N. PARKER. 1999. Application of nearest-neighbor and cluster analysis in pharmaceutical lead discovery. J. Chem. Inf. Comput. Sci. **39**: 21–7.

186 TAYLOR, R. D., P. J. JEWSBURY AND J. W. ESSEX. 2003. FDS: flexible ligand and receptor docking with a continuum solvent model and soft-core energy function. J. Comput. Chem. **24**: 1637–56.

187 TAYLOR, R. 1995. Simulation of experimental design strategies for screening random compounds as potential new drugs and agrochemicals. J. Chem. Inf. Comput. Sci. **35**: 59–67.

188 TODESCHINI, R., M. LASAGNI AND E. MARENGO. 1994. New molecular descriptors for 2D and 3D structures. J. Chemometrics **8**: 263–72.

189 TODESCHINI, R. AND V. CONSONNI. 2000. *Handbook of Molecular Descriptors.* Wiley-VCH, Weinheim.

190 TOLEDO-SHERMAN, L., E. DERETEY, J. J. SLON-USKIEWIECZ, et al. 2005. Frontal affinity chromatography with MS detection of EphB2 tyrosine kinase receptor. 2. Identification of small-molecule inhibitors via coupling with virtual screening. J. Med. Chem. **48**: 3221–30.

191 VANGREVELINGHE, E., K. ZIMMERMANN, J. SCHOEPFER, R. PORTMANN, D. FABBRO AND P. FURET. 2003. Discovery of a potent and selective protein kinase CK2 inhibitor by high-throughput docking. J. Med. Chem. **46**: 2656–62.

192 VARADY, J., X. WU, X. FANG, J. MIN, Z. HU, B. LEVANT AND S. WANG. 2003. Molecular modeling of the three-dimensional structure of dopamine 3 (D_3) subtype receptor: discovery of novel and potent D_3 ligands through a hybrid pharmacophore- and structure-based database searching approach. J. Med. Chem. **46**: 4377–92.

193 VEBER, D. F., S. R. JOHNSON, H.-Y. CHENG, B. R. SMITH, K. W. WARD AND K. D. KOPPLE. 2002. Molecular properties that influence the oral bioavailability of drug candidates. J. Med. Chem. **45**: 2615–23.

194 VEDANI, A., M. DOBLER AND M. A. LILL. 2005. Combining protein modeling and 6D-QSAR – simulating the binding of structurally diverse ligands to the estrogen receptor. J. Med. Chem. **48**: 3700–3.

195 VEDANI, A., M. DOBLER AND P. ZBINDEN. 1998. Quasi-atomistic receptor surface models: a bridge between 3-D QSAR and receptor modeling. J. Am. Chem. Soc. **120**: 4471–7.

196 VERDONK, M. L., J. C. COLE, M. J. HARTSHORN, C. W. MURRAY AND R. D. TAYLOR. 2003. Improved protein–ligand docking using GOLD. Proteins Struct. Func. Genet. **52**: 609–23.

197 VERDONK, M. L., V. BERDINI, M. J. HARTSHORN, W. T. M. MOOIJ, C. W. MURRAY, R. D. TAYLOR AND P. WATSON. 2004. Virtual screening using protein–ligand docking: Avoiding artificial enrichment. J. Chem. Inf. Comput. Sci. **44**: 793–806.

198 VERKMAN, A. S. 2004. Drug discovery in academia. Am. J. Physiol. Cell Physiol. **286**: C465–74.

199 VIGERS, G. P. A. AND J. P. RIZZI. 2004. Multiple active site corrections for docking and virtual screening. J. Med. Chem. **47**: 80–9.

200 WALTERS, W. P., M. T. STAHL AND M. A. MURCKO. 1998. Virtual screening – an overview. Drug Discov. Today **3**: 160–78.

201 WANG, J., L. LAI AND Y. TANG. 1999. Structural features of toxic chemicals for specific toxicity. J. Chem. Inf. Comput. Sci. **39**: 1173–89.

202 WANG, R., L. LAI AND S. WANG. 2002. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. J. Comput. Aided Mol. Des. **16**: 11–26.

203 WANG, R., Y. LU AND S. WANG. 2003. Comparative evaluation of 11 scoring functions for molecular docking. J. Med. Chem. **46**: 2287–303.

204 WANG, R., Y. LU, X. FANG AND S. WANG. 2004. An extensive test of 14 scoring functions using the PDBbind refined set of 800 protein–ligand complexes. J. Chem. Inf. Comput. Sci. **44**: 2114–25.

205 WANG, R. AND S. WANG. 2001. How does consensus scoring work for virtual library screening? An idealized computer experiment. J. Chem. Inf. Comput. Sci. **41**: 1422–6.

206 WARD, JOE H., J. 1963. Hierarchical grouping to optimize an objective function. J. Am. Stat. Ass. **58**: 236–44.

207 WATANABE, K., H. NAKACAWA AND S. TSURUFUJI. 1986. A new sensitive fluorometric method for measurement of plasma exudation in the inflammatory skin reaction. J. Pharmacol. Methods **15**: 255–61.

208 WEBER, A., A. TECKENTRUP AND H. BRIEM. 2002. Flexsim-R: a virtual affinity fingerprint descriptor to calculate similarities of functional groups. J. Comput. Aided Mol. Des. **16**: 903–16.

209 WEININGER, D., A. WEININGER AND J. L. WEININGER. 1989. Algorithm for generation of unique SMILES notation. J. Chem. Inf. Comput. Sci. **29**: 97–101.

210 WEININGER, D. 1988. SMILES, a chemical language for information szstems. 1. Introduction to methodology and encoding rules. J. Chem. Inf. Comput. Sci. **28**: 31–6.

211 WILLETT, P., J. M. BARNARD AND G. M. DOWNS. 1998. Chemical similarity searching. J. Chem. Inf. Comput. Sci. **38**: 983–96.

212 WILLETT, P., V. WINTERMAN AND D. BAWDEN. 1986. Implementation of nearest-neighbor searching in an online chemical structure search. J. Chem. Inf. Comput. Sci. **26**: 36–41.

213 WILLETT, P., V. WINTERMAN AND D. BAWDEN. 1986. Implementation of nonhierarchic cluster analysis methods in chemical information systems: selection of compounds for biological testing and clustering substructure search output. J. Chem. Inf. Comput. Sci. **26**: 109–18.

214 WOLBER, G. AND T. LANGER. 2005. LigandScout: 3-D pharmacophores derived from protein-bound ligands and their use as virtual screening filters. J. Chem. Inf. Model. **45**: 160–9.

215 WYSS, P. C., P. GERBER, P. G. HARTMAN, C. HUBSCHWERLEN, H. LOCHER, H.-P. MARTY AND M. STAHL. 2003. Novel dihydrofolate reductase inhibitors. Structure-based versus diversity-based library design and high-throughput synthesis and screening. J. Med. Chem. **46**: 2304–12.

216 XIE, D., A. TROPSHA AND T. SCHLICK. 2000. An efficient projection protocol for chemical databases: singular value decomposition combined with truncated-Netwon minimisation. J. Chem. Inf. Comput. Sci. **40**: 167–77.

217 XING, L., E. HODGKIN, Q. LIU AND D. SEDLOCK. 2004. Evaluation and application of multiple scoring functions for a virtual screening experiment. J. Comput. Aided Mol. Des. **18**: 333–44.

218 XUE, L., F. L. STAHURA, J. W. GODDEN AND J. BAJORATH. 2001. Mini-fingerprints detect similar activity of receptor ligands previously recognized only by three-dimensional pharmacophore-based methods. J. Chem. Inf. Comput. Sci. **41**: 394–401.

219 XUE, L., J. W. GODDEN AND J. BAJORATH. 1999. Database searching for compounds with similar biological activity using short binary bit string representation of molecules. J. Chem. Inf. Comput. Sci. **39**: 881–6.

220 XUE, L., J. W. GODDEN AND J. BAJORATH. 2000. Evaluation of descriptors and mini-fingerprints for the identification of molecules with similar activity. J. Chem. Inf. Comput. Sci. **40**: 1227–34.

221 YOUNG, S. S. AND D. M. HAWKINS. 1995. Analysis of a 2^9 full factorial chemical library. J. Med. Chem. **38**: 2784–8.

19
Efficient Strategies for Lead Optimization by Simultaneously Addressing Affinity, Selectivity and Pharmacokinetic Parameters

Karl-Heinz Baringhaus and Hans Matter

1 Introduction

The increasing pressure on efficiency and cost of research in the pharmaceutical industry has caused a technological paradigm shift [1, 2] in order to bring promising candidate molecules earlier to the market [3, 4]. The total research and development costs for a novel compound to enter this market have recently been estimated as US$600–800 million [5, 6], while continuously increasing expenses are mainly attributed to the high attrition rate in later development phases. The major reason for failure in phase II and III of clinical trials is inadequate understanding of pharmacokinetic behavior of drugs [7–9] and what constitutes a pharmacokinetic profile for candidate drugs. This again underscores the necessity to improve on the success rates in order to maintain economic viability of the pharmaceutical drug discovery process on longer terms.

Several technological advances [10] along the drug discovery value chain have entered the field and become indispensable tools for identifying novel lead compounds. However, the advent of technologies like combinatorial chemistry [11, 12], automated synthesis technologies and high-throughput screening (HTS) [13] also caused an exponential increase in the number of single data points for analysis, while it is debatable to what extent these technologies contributed to the launch of new chemical entities (NCEs) to the market [2, 3, 14, 15]. The current challenges in drug discovery are related to increased regulatory hurdles, more effective integration of technological advances, the extraction of relevant information and knowledge from primary data to support data-driven decisions, and – at the same time – the cost pressure, prompting for increased efficiency and lower attrition rates.

Improving on the low success rates requires a balanced compound progression driven by clearly defined knowledge-based decisions to advance or discontinue particular lead series as early as possible. Any failure to identify promising lead series has severe implications on time and resources within a disease-related program in a pharmaceutical company. A clear process,

Bioinformatics – From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

and rigorous and relevant metrics for compound progression are essential here [2, 16]. This requires early attention to stringent quality criteria for compound series, reflecting the complex multitude of biological, pharmacological, pharmacokinetic and physicochemical parameters that a project team has identified as key drivers for the chemical optimization and for candidate selection.

Although, at first glance, the stage of lead optimization is not very spectacular, the efficient conversion of molecules with promising biological activity into viable drug candidates fulfilling a multitude of requirements appears to be one of the most challenging steps [17, 18] with a very high impact on the successful continuation of efficient drug discovery programs [2]. Conversion of a biologically active chemical compound into an effective and safe drug adds substantial value in the drug discovery process. Consequently, the improvement of a compound profile toward a clinical candidate is one of the essential skills in integrated drug discovery teams. Those candidate requirements include not only desirable selectivity against related or diverse "antitargets", but also favorable physicochemical and pharmacokinetic properties, leading to oral administration and an acceptable half-life of the final candidate. Hence, to arrive at candidates with suitable pharmacodynamic and pharmacokinetic properties requires a simultaneous optimization of multiple parameters in carefully planned iterations. Here, it is interesting to note that recent comparisons of molecular properties of launched drugs revealed only limited differences compared to the original lead compound as starting point [19].

The necessity to shorten the discovery process has prompted for an early integration of pharmacokinetic and drug development efforts to rapidly identify those molecules that are unlikely to become drugs and those lead series with significantly lower chemical optimization potential. It is of great importance to initiate lead optimization programs only for those molecules that intrinsically have the potential to be converted into drugs. Hence, this lead optimization stage has entered into a new phase of complexity caused by advances in modern technologies like protein crystallography, assay technology, absorption, distribution, metabolism and excretion (ADME) assays, medicinal chemistry automation, etc. Although rational approaches during this phase to manipulate molecules are typically guided by quantitative structure–activity relationships (QSARs) and structure-based design, it now becomes a tight interplay between various disciplines: medicinal chemistry, structural biology, pharmacology and pharmacokinetics. It is vital for success to conceive lead optimization as simultaneous multidimensional optimization rather than to address one parameter at a time (Figure 1).

This chapter focuses on a discussion of novel strategies, processes and computational tools with an impact on improving the poor industry success

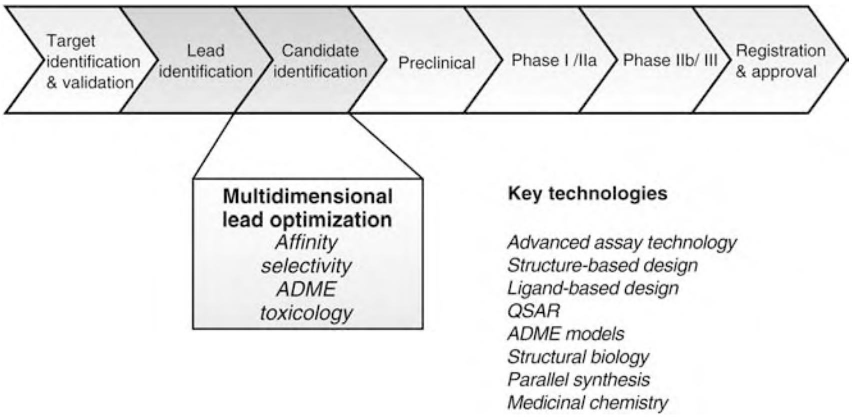


Figure 1 Drug discovery value chain, and selected key technologies and activities [2]

rates during the chemical optimization phase of lead series. The ultimate goal of any chemical optimization program is to convert quality lead structures into high-value drug candidate molecules prior to clinical development (Figure 1). First, the starting point for lead optimization is analyzed, i.e. the origin of lead and current selection criteria for the promising series. Knowledge-based decisions are essential at this stage to circumvent time and resource investments for later stages. The emphasis of subsequent sections is to discuss several components in this lead optimization process toward refinement of series with improved drug-like properties. This includes a summary of the understanding and predictive models for biological affinity, selectivity versus closely related targets as well as clearly defined antitargets and pharmacokinetic problems.

Traditionally, in this optimization process, the affinity is optimized first, while pharmacokinetic and ADME issues are investigated later in this process [20, 21]. This approach, however, showed only limited success, as optimizing for affinity only can result in chemical classes, where subsequent optimization for ADME properties becomes difficult, if not impossible. The efficiency of the drug discovery process is expected to improve if both aspects are considered simultaneously (Figure 2). Hence, we discuss different approaches and case studies on lead optimization using such a tight integration of different parameters by means of simultaneous optimization of a variety of relevant molecular properties, sometimes referred to as *multidimensional optimization* (MDO) [2, 16, 22]. Consequently, we will use this term throughout the following discussion. This multidimensional optimization requires a clear risk assessment prior to initialization of a resource-intensive research program to optimize compounds toward a balance of such a multitude of properties [23]. General considerations for multidimensional lead optimization are discussed

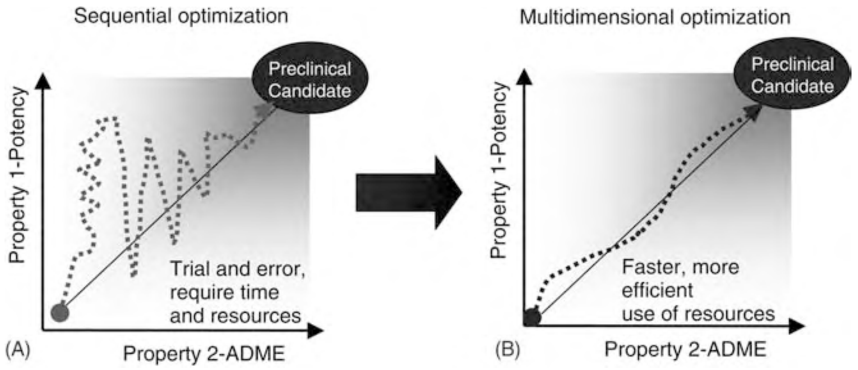


Figure 2 Different drug discovery strategies: (A) Sequential optimization (historic approach) – first optimization of affinity, while ADME properties are treated at a later stage. (B) Multidimensional scenario – combined optimization of affinity and ADME properties, simultaneously monitor changes in relevant properties [20]

with an emphasis on computational approaches to assist in this simultaneous task. Data integration is mandatory for extraction of knowledge and reuse in following iterations in the design cycle, while *in silico* approaches to estimate ADME parameters are crucial to focus only on promising molecules to address the increased complexity in lead optimization. These concepts will be illustrated using literature and in-house examples (Figure 2).

2 The Origin of Lead Structures

This section summarizes strategies to identify lead structures with a particular emphasis on modern technologies complementing HTS in the pharmaceutical industry, while Bleicher and coworkers discuss some aspects in detail [22], and provide a summary of the individual processes and interfaces between disciplines involved within a pharmaceutical setting [16].

The success of many drug development projects seems to be fundamentally limited by the nature of the target. Experience in medicinal chemistry and historic drug discovery programs suggests that small-molecule organic compounds can modulate more readily some privileged protein target classes. Hence, the selection of tractable targets in addition to careful selection following therapeutic requirements is essential to guide drug discovery at an early stage. All drugs that are currently on the market are estimated to modulate less than 500 targets, like nucleic acids, enzymes, G-protein-coupled receptors (GPCRs) or ion channels [9]. Consequently, the “druggability” of protein targets including small-molecule binding sites has been discussed [24] following the completion of the Human Genome Project, with an attempt to estimate a reasonable number of targets that indeed provide an opportunity for future therapeutic intervention. While earlier estimations discuss a total of 5000–10 000 potential targets according to the number of disease-related genes [9], a focus on properties of orally bioavailable, small drug molecules that could potentially interact with those targets results in a lower number of targets (600–1500) [24]. Those physicochemical properties of "drug-like" molecules include the well-established "Rule of Five" [25] or related knowledge-derived guidelines [26]. Those summarize simple, intuitive parameters for medicinal chemists to give a warning for compounds that are unlikely to be orally absorbed via the passive intestinal route. Hence, the increasing understanding of drug-like properties might additionally refine our understanding of complementary properties for drug targets.

Hit finding for a selected target typically starts with a collection of actives from HTS [13, 27, 28], which today is the most widely applicable technology, while quality of results is critically dependent on assay type and quality [16]. In addition, recent years have witnessed an emerging number of complementary biophysical and *in silico* hit-finding approaches like fragment-based SAR-by-nuclear magnetic resonance [29–31], surface plasmon resonance binding [32], high-throughput X-ray crystallography [33, 34], protein structure-directed *de novo* design [35], and structure- and ligand-based library design and virtual screening [36, 37].

The costs associated with hit identification using diverse assay systems and technologies as well as, alternatively, biophysical methods and approaches, are regarded as minor compared to the finally required clinical development of novel candidate molecules [38]. The validation and exploration around lead series is significantly more resource-intensive and dependent upon the selection criteria to progress on promising series. Hence, the series of hits are thoroughly evaluated and validated as early as possible for collecting as much information on individual structures and entire structure classes. Any systematic method to evaluate and validate results from hit finding requires a clear understanding of which terms and criteria to monitor in order to decide upon the fate of a series [2, 16].

Several criteria for selecting promising lead structures and tailoring focused libraries have been proposed in the literature. The early awareness of liabilities within a lead series with respect to the desirable compound profile in terms of selectivity, physicochemical and ADME properties is important for choosing the series with potential for starting a successful lead optimization program; however, none of these knowledge-driven decisions can prevent unpredictable problems like animal toxicity and others. Generally speaking, it is the overall characteristics of a particular compound class, considering a multitude of properties including synthetic feasibility and patentability, which make it attractive for starting a lead optimization program. This potential, however, has to be systematically explored within the frame of a limited synthetic program dedicated to exploring the SAR as well as obtaining as much information on key drivers in order to identify those series with improved drug-like properties.

Recent comparisons of physicochemical properties of lead structures to drug molecules also enhanced the understanding of which criteria to apply for selecting a promising series. Those studies were reported by Oprea and coworkers [39–41] based on a data set of 96 lead-drug pairs and a variety of calculated physicochemical properties. Lead structures tend to exhibit lower molecular weight, lower number of rings and rotatable bonds and lower log P, suggesting that the subsequent process of lead optimization in medicinal chemistry tends to add complexity, mainly by means of less directional hydrophobic interactions.

The impact of molecular complexity on the probability to detect hits was studied by Hann and coworkers [42] on the basis of statistical arguments. Using simple models of ligand–receptor interactions, the probability of useful binding events of increasing complexities (number of potential protein–ligand interactions) was estimated. Low-complexity compounds have an increased probability of being detected as hits in screening and thus might offer better starting points for drug discovery. If systems become more complex, the chance of observing a useful interaction for a randomly picked ligand falls dramatically, based on statistical reasons.

One possible route to increase binding affinity of low-complexity fragments is to link two fragments binding to different subpockets. From an analysis of high-resolution X-ray data for fragments and corresponding larger molecules, the loss of rigid-body translational and rotational entropy, which forms a significant barrier of protein–ligand recognition, was estimated as 15–20 kJ mol^{-1}, i.e. 3 orders of magnitude in affinity at 298 K [43].

These studies consistently suggest that less-complex lead structures are better to detect and exhibit a favorable optimization potential, which might influence attempts to design more “lead-like” [44] compound libraries for screening. They have also lead to alternative lead-finding strategies directed toward the initial discovery of small fragments as starting points for optimization [45]. Recent examples include the discovery of DNA gyrase inhibitors by means of three-dimensional (3-D) structure-based fragment screening combined with biophysical assays and protein structure-based optimization [46] and the discovery of inosine 5′-monophosphate dehydrogenase (IMPDH) inhibitors [47] from a virtual screening protocol tailored to discover small, alternative warheads to known phenyloxazole anilines.

The combination of alternative screening approaches, virtual screening and parallel medicinal chemistry in combination with an early profiling on the multitude of relevant compound properties will hopefully generate an improved basis for proper decisions about which promising lead series to take forward.

3 Optimization for Affinity and Selectivity

3.1 Lead Optimization as a Challenge in Drug Discovery

This section summarizes several approaches with and without knowledge of the target protein 3-D structure to optimize lead structures for affinity and selectivity as one prerequisite for multidimensional optimization. The lead optimization phase to efficiently convert the lead structures with promising biological properties into clean drug candidates fulfilling a multitude of criteria is a challenging task [17, 18] with a high impact on the successful continuation of efficient drug discovery programs [2]. Consequently, the improvement of a compound profile toward a clinical candidate is one of the essential skills in integrated project teams. Those candidate requirements include not only desirable selectivity against related or diverse “antitargets”, but also favorable physicochemical and pharmacokinetic properties, leading to oral administration and an acceptable half-life of the final candidate. To arrive at candidates with suitable pharmacodynamic and pharmacokinetic properties thus requires a simultaneous optimization of multiple parameters in carefully planned iterations.

Due to the lower dimensionality of the ADME space, ADME properties should be easier to predict than biological receptor affinity [48], although in practice this is more difficult, as many experimental screens for ADME properties are multi-mechanism rather than single mechanism systems. In contrast, biological assays for the majority of pharmacological targets are typical single-mechanism systems, for which computational models to correlate structural descriptors are easier to develop and resulting predictions tend to be more robust. Many experimental screens for these ADME properties include multiple-mechanism systems, however. Computational models, on the other hand, for data with multiple underlying mechanisms tend to get worse if more data for structurally more diverse compounds are experimentally obtained and included into the training set. This is mainly due to the fact that the increase of assay data relates to an increase of underlying mechanisms on which those data have been obtained and the noise level rises for each individual mechanistic component [48]. For smaller, structurally and, thus probably, mechanistically homogeneous data sets, acceptable correlations are obtained, while the ability of descriptors to capture a more diverse experimental data set is limited. Although the same descriptors might still have some statistical significance and thus explain global trends for less homogeneous data, their predictivity is often too low for a valid *in silico* prediction. This only offers the possibility to construct statistically significant rules or filters based on descriptors and property distributions and ranges. Hence, it is mandatory to carry out high-quality single-mechanism ADME experimental data and use them to derive single-mechanism models [48]. In contrast, assays and models to correlate structural properties to biological activity for a biomolecular target are mainly based on a single mechanism, i.e. favorable protein–ligand interactions upon the molecular recognition event.

Although the present chapter does not summarize the pharmacophore [49, 50] and 2-D/3-D-QSAR [51–55] approaches toward ligand-based optimization in the absence of any 3-D receptor structure, these methods are extremely important in today's lead optimization settings, as for the majority of targets it might be difficult to obtain relevant X-ray structures of protein–ligand complexes. This is true for membrane-bound proteins, like ion channels, GPCRs and others. Hence, validated methods in ligand-based design to extract knowledge from molecules in a series, build statistical models and use them for further design are useful. The focus here is on the use of 3-D protein structure information *per se* and of the tight integration of ligand- and structure-based design approaches toward a more reliable affinity prediction and understanding of selectivity. However, both ligand- and structure-based design approaches require the close monitoring of ADME and physicochemical properties of ligand series for successful optimization.

3.2 Use and Limitation of Structure-based Design Approaches

Over the past few years, there has been an exponential increase in the number of available 3-D structures in the public domain [56, 57], which is increasingly being used within the drug discovery process in the lead-finding and lead optimization phase. However, there are still other factors that add up to the complexity of guided lead optimization by structure-based design [58–62]. The reality of protein–ligand interactions at the molecular level is still far too complex to provide a correct *in silico* approach for accurate affinity prediction, either based on the knowledge of the target protein 3-D structure or even without that information. The effect of entropy and the dielectric constant are only two examples that are controversially discussed within the literature. Other challenging factors in structure-based design are the existence of multiple ligand-binding modes [63, 64], the accessibility of conformational states for both ligand and receptors [65, 66], the influence of structurally conserved water [67], pH [68], and others. One important point to address is the well-documented protein flexibility [65] as a direct consequence of the chemical properties of amino acids. Furthermore, it is of critical importance in the structure-based design process to reflect the limitations of the use of X-ray protein structures [69], which also is partially due to significant errors in some reported structures [70, 71] and experimental limitations that are difficult to overcome. These complicating influences turn the reality of lead optimization for affinity into a difficult task.

Some important reviews summarized the current status of our understanding of protein–ligand interactions [72, 73]. This knowledge prompted some groups to derive scoring functions for guiding flexible docking algorithms and rank-ordering series of docked molecules. Several publications indicate the substantial progress made in both fields of docking and scoring, respectively. A variety of methods exist to estimate how strongly a ligand interacts with a protein-binding site, while today's existing functions belong to three main categories: force field-based methods, empirical scoring functions and knowledge-based methods [62, 73–77]. In an intersection-based consensus scoring approach, Charifson and coworkers [78] and Clark and coworkers [79] combined a range of different functions to rank protein–ligand geometries, which resulted in an increased performance with respect to hit rates. The first paper [78] also discusses one of the possible limitations for earlier scoring functions, i.e. that those are derived from X-ray structures of extremely potent ligands, while information about less-active analogs is often not available.

3.3 Integration of Ligand- and Structure-based Design Concepts

The global scoring functions described above are derived from a more or less balanced collection of protein–ligand complexes from the public domain, leading to empirical functions that might be able to globally provide some guidelines on whether a compound could potentially bind to a particular binding site. However, their accuracy in ranking compounds on the basis of experimental binding affinities is typically limited. This caused studies to tailor scoring functions in lead optimization stages using a narrow training set for only the protein target of interest. In addition, this need in lead optimization prompted others toward a tight integration of ligand- and structure-based design approaches to model and understand biological affinities by means of 3-D-QSAR methods, which are based on an alignment rule derived from reliable docking modes.

Gohlke and coworkers developed the AFMoC procedure [80] to derive tailored scoring functions for protein–ligand complexes based on DrugScore [81] knowledge-based pair-potentials, which are adapted to a single protein-binding site by incorporating ligand information (Figure 3). The formalism is similar to comparative molecular field analysis (CoMFA) [82, 83], while the fields in their approach originate from the protein environment and not from the ligands alone. A regular-spaced grid is placed into the binding site and knowledge-based pair-potentials between protein atoms and ligand

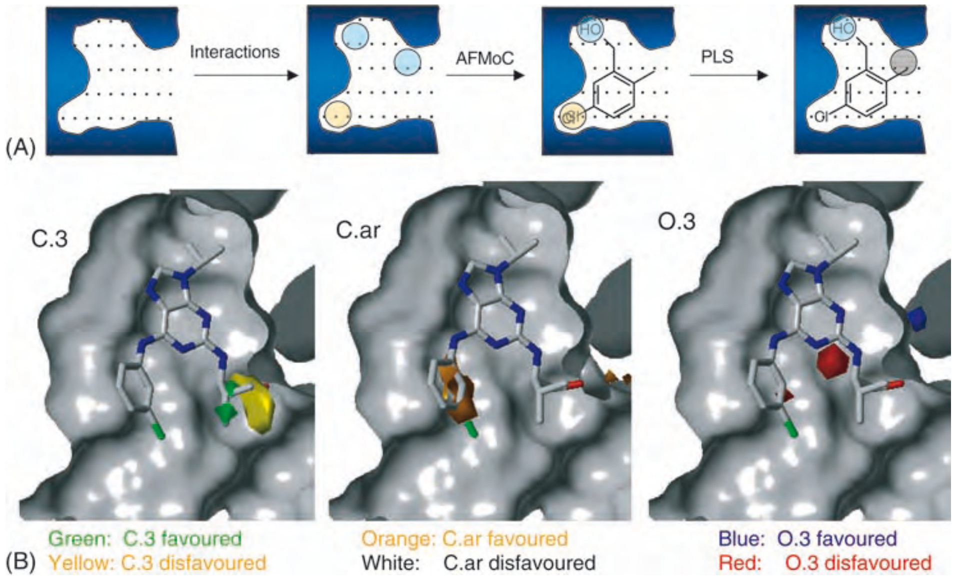


Figure 3 (A) Deriving a tailored scoring function using AFMoC [80], which integrates protein- and ligand-derived descriptors and produces a statistical model on protein–ligand interactions relevant for affinity. On a predefined grid, favorable protein interactions with ligand atom types are computed. These are adapted by incorporating actual information from a series of docked or crystallized ligands. A statistical analysis led to a model with favorable and unfavorable contributions. (B) AFMoC interpretation for 86 substituted purines as CDK-2 inhibitors. The underlying PLS model (q^2: 0.521, r^2: 0.800, 5 PLS components) indicates favorable and unfavorable regions for atom types, as shown for C.3, C.ar and O.3, and the CDK-2/purvalanol A complex (PDB 1CKP). The CDK-2 binding site is represented as a MOLCAD solvent-accessible surface.

atom probes are mapped onto the grid intersections resulting in "potential fields". In a partial least square (PLS) [84] analysis, these atom-type specific interaction fields are correlated to the actual binding affinities of the docked or experimentally known ligands, resulting in individual weighting factors for each field value. They described significant improvements of the predictive power for affinity prediction compared to global knowledge-based potentials by considering a sample set of only 15 known training ligands.

The conceptually similar COMBINE approach was developed by Wade and coworkers [85] on the basis of the analysis of force-field energy contributions per amino acid to describe interaction differences to a congeneric set of ligands. The resulting predictive regression equations were also reported for applications in structure-based design projects [86–88].

Other approaches are based on tailor-made empirical functions, consisting of additional descriptors and weights for these, plus the known terms using appropriate statistical methods based on a training set of several ligands exhibiting a range of biological affinities to only one protein cavity. Approaches to understand the SAR in chemotypes by a determination of the importance of weights of different coefficients of individual terms in scoring functions are described by Rognan and coworkers [89] and Murray and coworkers [90], both using the ChemScore global scoring function [91] as a start.

Structure-based design is focused on understanding protein–ligand interactions but does not always lead to predictive models for ligand series. In contrast, 3-D-QSAR with acceptable statistics does not always reflect topological features of the protein structure, as those are not necessarily built using alignment rules that reflect the bioactive conformation. Hence, several groups have successfully combined both approaches. On the basis of the protein X-ray structure and a series of analogs with a potentially similar binding mode, consistent and highly predictive 3-D-QSAR models were derived, which could be mapped back to the protein topology. This leads to a better understanding of important protein–ligand interactions, and provides guidelines for ligand design and a predictive model for scoring novel synthetic candidates. Docking calculations based on already available 3-D structures often might result in an alignment for all other compounds by superimposing them onto the template and relaxing them within the cavity for consistent 3-D-QSAR models.

These receptor-based 3-D-QSAR models represent a strategy to integrate ligand- and structure-based design approaches. CoMFA [51, 82, 83] and comparative molecular similarity index analysis (CoMSIA) [92] are used to derive relationships between molecular property fields and biological activities. Electrostatic and steric interaction energies are computed between each ligand and a probe atom located on predefined grid points for CoMFA, while for CoMSIA those interaction fields are replaced by fields based on similarity indices between probe atoms and each molecule. The PLS [84] method is used to derive a linear relationship, while cross-validation [93] is used to check for consistency and predictiveness. The resulting contour maps from 3-D-QSAR models enhance the understanding of electrostatic, hydrophobic and steric requirements for ligand binding, guiding the design of novel inhibitors to those regions where structural variations altering steric or electrostatic fields reveal a significant correlation to biological properties. The 3-D-QSAR results then allow focusing on those regions where steric, electronic or hydrophobic effects play a dominant role in ligand–receptor interactions.

One of the earlier and most influential applications of receptor-based CoMFA studies was carried out by Marshall and coworkers on a series of HIV protease inhibitors [94, 95]. The bound conformation of several ligands was known from X-ray crystallographic studies, and was shown to provide the most consistent and predictive QSAR models, while the authors also pointed out that conformational energy and entropic effects are not adequately included within these data sets. Hydrophobic field types used within CoMSIA [92] or HINT [96] might be partially useful to overcome the second limitation. Other successful applications include estrogen receptor ligands [97], acetyl cholinesterase inhibitors [98], and (from the authors' own work) receptor-based QSAR models for matrix metalloproteinase [99, 100], CDK-2 [101] and factor Xa inhibitors [102]. These models collectively found applications in lead optimization projects, as they provide an enhanced understanding of important effects in protein–ligand interactions and reliable affinity predictions.

3.4 The Selectivity Challenge from the Ligand's Perspective

Protein target families are the fundamentals to build a framework for matching biological motifs with chemical scaffolds. Many structurally related targets find implications in different therapeutic areas, while exhibiting similar protein–ligand interactions. A key issue for this interdisciplinary field is how to generate and efficiently use this information to guide drug design across targets [2]. The knowledge of structurally related proteins for a particular target is of interest in early drug discovery stages to identify leads by analogy within a target family [2, 22] from focused libraries enriched with privileged motifs of importance for that family [103] or by annotation-based similarity searching across target families [104, 105]. This latter approach allows us to perform similarity searching to identify appropriate ligands for new targets within the same target family by linking chemical structure databases, biological target sequences and structure activity data (screening results).

However, selectivity toward only a single biological target is an essential requirement for drug candidates to minimize side-effects. Hence, it is mandatory to involve selectivity considerations early in the drug discovery process, and to monitor experimentally and by *in silico* approaches closely related or structurally unrelated "antitargets" for the potential drug candidate. The following section will briefly summarize methods to take this requirement into account from both the perspective of the ligand and the protein 3-D target structures, if available.

While most chemistry-driven approaches rely on trial and error, often 2- and 3-D-QSAR approaches have been used to correlate biological affinities against both target proteins in order to identify those structural determinants or favorable spatial regions around the ligands gaining selectivity on the target enzymes. This approach requires a series of biologically characterized chemical analogs, which are typically available at this stage within the drug discovery value chain.

Some earlier examples on the use of 3-D-QSAR to understand selectivity differences include the work of Wong and coworkers to unveil structural requirements for selective binding of ligands to the diazepam-insensitive (DI) benzodiazepine receptor [106]. The chemical interpretation of the resulting CoMFA model on a larger series of 1,4-benzodiazepines highlighted the marked influence of particular substitutions in only a few scaffold positions on the ratio of affinities against the DI and diazepam-sensitive (DS) benzodiazepin receptor-binding site. Interestingly, individual 3-D-QSAR models have been derived for binding affinities against both individual sites as well as against the ratio, defined as $K_\mathrm{i}(\mathrm{DI})/K_\mathrm{i}(\mathrm{DS})$, as an additional dependent variable for statistical analysis. The use of K_i values for this mathematical transformation is preferable due to their independence on concentration effects, while IC_{50} values for closely related targets or sites at one receptor are often practically determined at very similar conditions, thus also allowing the use of similar ratios as dependent variables for statistics. However, PLS can also be used for two (or more) dependent Y-variables in only one model [84].

A similar approach has been taken to explain selectivity differences between the matrix metalloproteinases MMP-8 and MMP-3 [100] and the serine proteases factor Xa and thrombin [107]. However, in both cases, the availability of experimental 3-D protein structures also allowed the interpretation of selectivity differences by careful inspection of protein-binding site requirements and additional statistical models tailored to highlight key differences in potential protein–ligand interactions from binding sites only [100, 101, 107, 108]. In both cases, the selectivity differences from the analysis of ligand series nicely correspond to binding site requirements. Furthermore, the availability of 3-D structures for both the target and the closely related “antitarget” might lead to the use of structure-based design approaches to qualitatively identify amino acid differences, which might allow for a more focused interaction toward only one target. The combined use of both types of information greatly enhanced the understanding of selectivity requirements for these series of target enzymes.

Other quantitative approaches to address selectivity rely on generating and consistently interpreting individual statistical models, which again provide very detailed information on desirable scaffold substitution patterns and spatial areas around superimposed ligands, which allow them to selectively interact only with one binding site. There are multiple interesting applications of this concept in the literature (e.g. Refs. [109–112]).

3.5 Selectivity Approaches Considering Binding Site Topologies

The availability of protein 3-D structures considerably simplifies the search for selective ligands, as these structures allow the comparative analysis of favorable protein–ligand interactions with a particular focus on those that are only possible in one member of a protein family. There are several computational tools that assist in an unbiased characterization of protein–ligand interactions, e.g. the force field-based approach GRID [113], and the knowledge-based approaches SuperStar [114, 115] and DrugScore [81, 116]. In the recent consensus principal component analysis (GRID/CPCA) approach [108], the GRID descriptors derived for multiple protein binding sites are analyzed using CPCA [117] as a statistical method, which allows the identification of possible modifications in potential ligands in a straightforward way to improve their selectivity toward a chosen target. This statistical analysis evaluates the relative importance of individual molecular interaction fields from particular probe atoms for the final PCA model. As the input data matrix is structured in chemically meaningful blocks, hierarchical and consensus multivariate analysis tools like CPCA provide information about the relative importance of individual blocks on the stage of an intermediate level of the analysis, while the final model is identical to a regular PCA. Plotting of the individual PCA and CPCA scores then allows visualization of selectivities between different members of the protein family, while the structural reason in terms of potential protein–ligand interactions that allows discrimination can be deduced from CPCA loadings plots and interactive variations thereof. Those "active plots" help to focus ligand designs toward binding site residues, which are essential for discriminating between subfamilies.

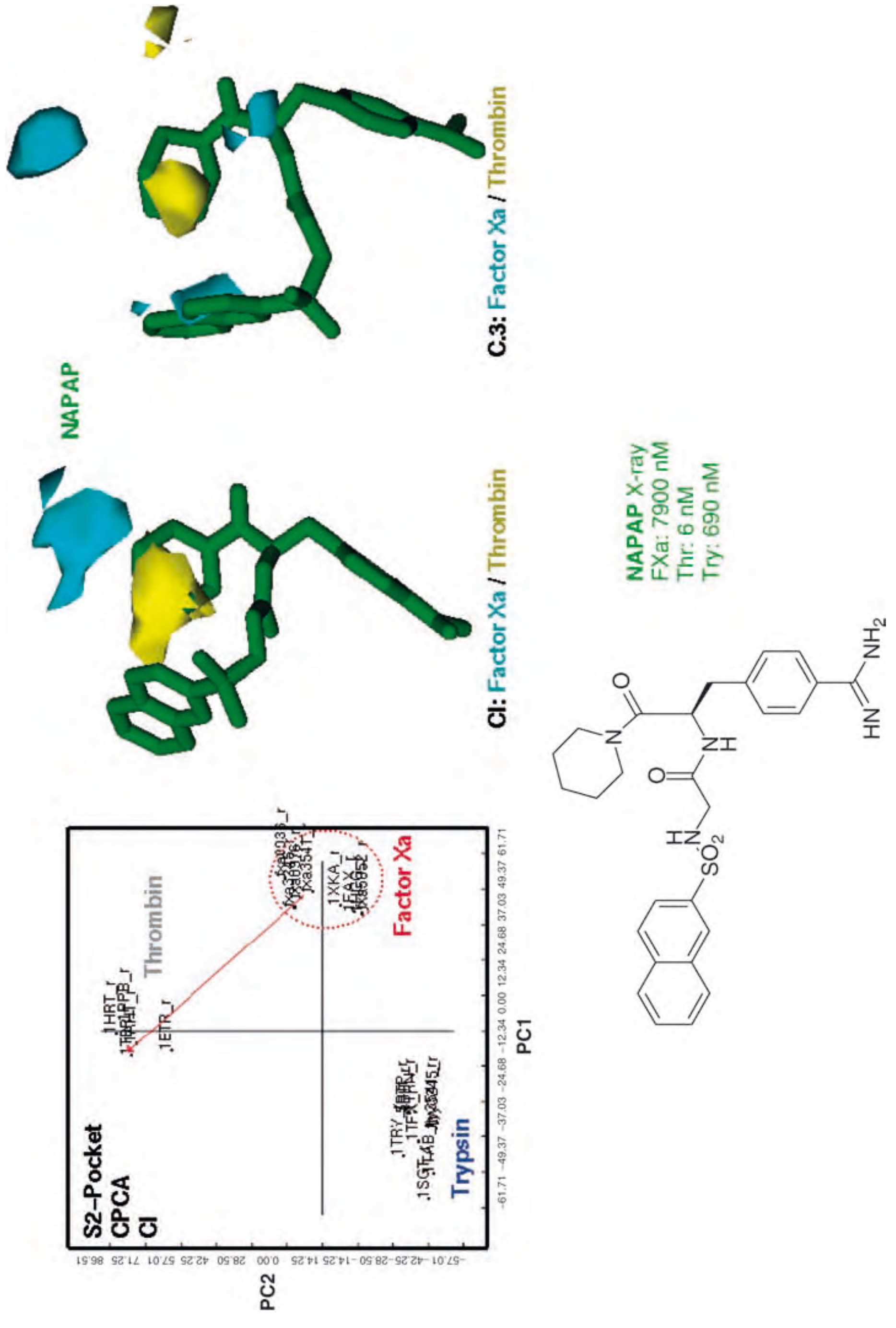
This approach was first applied toward an understanding of discriminating interactions in the serine proteases factor Xa, thrombin and trypsin [108], and provided selectivity information for all important serine protease subpockets, which are in agreement with experimental selectivities of typical protease inhibitors. This approach was complemented by a 3-D-QSAR selectivity analysis on a series of 3-amidinobenzyl-1H-indole-2-carboxamides [107], which points, from the viewpoint of the ligands, to similar main interactions driving selectivity between key enzymes in the blood coagulation cascade: factor Xa and thrombin. Other applications of GRID/CPCA include the interpretation of structural differences in human cytochromes P450, 2C8, 2C9, 2C18 and 2C19 [118] from homology models using the mammalian CYP2C5 X-ray structure [119], matrix metalloproteinases from X-ray and homology modeling [120], the classification of Eph Kinases [121], and a comparison of the binding characteristics of the family of lipid-binding proteins [122]. For these cases, the selectivity regions were in good agreement to available experimental information and inhibitor structure–activity relationships. Recent extensions include FLOGTV toward a simplified visualization of differences between related receptor sites based on a trend vector analysis [123].

Many structurally related protein targets find implications in different therapeutic areas, while exhibiting similar protein–ligand interactions. It is essential to efficiently use this information to guide drug design across targets. This encompasses both the knowledge about selectivity regions in binding sites, while, on the other hand, a general entrance by less-specific interactions into such target family is of particular interest for shaping a compound library at the lead-finding stage. To this end, GRID/CPCA was successfully applied to arrive at the "target family landscape" concept [101] to study features for general interactions to an entire target family and to identify determinants for interactions to only one member, which was of particular interest for broader sets of kinases [101] and serine proteases [107] with experimentally known 3-D crystal structures. The applications to major protein target families produced models in agreement with in-house ligand SAR models and thus provide guidelines toward rapid optimization in a library format for family-specific scaffolds.

To use a set of consistent descriptors for docking and selectivity studies, the replacement of the force field GRID by DrugScore-derived [81, 116] knowledge-based potentials was successfully investigated on the factor Xa/thrombin/trypsin problem [124] as well as for matrix metalloproteinases. It was shown that this change produces models of qualitatively similar interpretation. Predictive submodels for all protein pockets (S1, S2, S4) were obtained. Here, CPCA scores discriminate factor Xa from thrombin and trypsin on PC1 in Figure 4 (Cl probe for S2 submodel). Differential plots reveal the structural reason for this discrimination: the piperidinyl moiety in the thrombin selective inhibitor NAPAP can only favorably interact with this subsite in thrombin. The detailed interpretation for other pockets is consistent and in agreement with internal series.

Lapinsh and coworkers [125] developed the proteochemometrical strategy combining ligand and protein information for a comparative analysis of series of receptors and ligands. Proteochemometrics exploits affinity data for series of diverse organic ligands binding to different receptor subtypes, correlating it to descriptors and cross terms derived from the amino acid sequences of the receptors and the structures of the small ligands. Statistically valid models resulted in all cases with good external predictive ability, while evaluation of these models gave important insight into the mode of interactions of the GCPRs with their ligands. This method was successfully applied to other GPCR receptor subtype analyses without experimental knowledge of binding site topologies, i.e. melanocortin receptor subtypes [126], serotonin, dopamine, histamine, adrenergic receptor subtypes [127], and α_{1a}-, α_{1b}- and α_{1d} -adrenoceptors [125]. This approach combining information from ligands and receptors provided more detailed information about receptor–ligand interactions and determinants for receptor subtype selectivity than ligand-based QSAR studies on individual ligand series alone.

These presented examples of quantitative descriptions of protein–ligand interactions remain to be a very promising area to address affinity and the selectivity challenge in the future. Those approaches are collectively seen


S2–Pocket
CPCA
CI
Thrombin
Factor Xa
Trypsin
1HRT_r
1ETR_r
1XKA_r
1FAX_r
1TRY
1SGT
1TAB
PC1
PC2
-61.71 -49.37 -37.03 -24.68 -12.34 0.00 12.34 24.68 37.03 49.37 61.71
-57.01 -42.25 -28.50 -14.25 14.25 0.00 28.50 42.25 57.01 71.25 86.51
NAPAP
CI: Factor Xa / Thrombin
C.3: Factor Xa / Thrombin
NAPAP X-ray
FXa: 7900 nM
Thr: 6 nM
Try: 690 nM
O
N
H
SO2
HN
NH2

as interesting for a more integrated lead optimization on a simultaneous consideration of numerous balanced descriptors and models for optimization.

4 Addressing Pharmacokinetic Problems

4.1 Prediction of Physicochemical Properties

Pharmacokinetics and toxicity have been identified as important causes of costly late-stage failures in drug development. Hence, physicochemical as well as ADMET properties need to be fine-tuned even in the lead optimization phase. Recently developed *in silico* approaches will further increase model predictivity in this area to improve compound design and to focus on the most promising compounds only. A recent overview on ADME *in silico* models is given in Ref. [128].

The physical properties of a compound also determine its pharmacokinetic and metabolic behavior in the body. Poor biopharmaceutical properties are often linked to poor aqueous solubility, as summarized by Lipinski [129]. Thermodynamic solubility measurements consider the crystal packing energy of the solute as also its cavitation and solvation energy. The crystal packing energy accounts for disruption of the crystal to bring isolated molecules into the gas phase. The cavitation energy is the energy required to disrupt water for the creation of a cavity in which the solute is to be hosted. Finally, the solvation energy is the sum of favorable interactions between solvent and solute. Solubilization is largely kinetically driven, and the effects of crystal packing energy and polymorphic crystal forms are lost.

Currently available solubility models based on turbidimetry as well as nephelometry are not very predictive and are limited in their broad applicability, because they make use of training data from different laboratories determined under varying experimental conditions [130]. However, access to many aqueous solubilities measured under standardized conditions is expected to greatly improve currently available models.

←

Figure 4 DrugScore/CPCA for ligand selectivity in structure-based design. The binding sites of serine protease X-ray structures (factor Xa/thrombin/trypsin) are profiled using DrugScore for favorable interactions of six atom types (C.3, O.2, N.am, C.ar, O.3 and Cl). Individual CPCA models are derived for each subsite, allowing the identification of regions for discrimination. The interpretation for the S2 pocket and NAPAP as thrombin-selective inhibitor (PDB 1ETS) highlights yellow regions, where hydrophobic groups increase thrombin selectivity, while blue regions should favorable effect factor Xa selectivity. The interpretation is in good agreement with internal SAR data.

4.2 Prediction of ADME Properties

Poor intestinal absorption of a potential drug molecule can be related to poor physicochemical properties and/or poor membrane permeation. Poor membrane permeation could be due to low paracellular or transcellular permeability or the net result of efflux from transporter proteins including MDR1 [P-glycoprotein (P-gp)] or MRP proteins situated in the intestinal membrane. Cell lines with only one single efflux transporter are currently engineered for *in vitro* permeability assays to obtain suitable data for reliable QS models. In addition, efforts to gain deeper insight into P-gp and the ABC transporter on a structural basis are ongoing [131, 132].

Discrimination of efflux, active or passive transport is already feasible by suitable *in vitro* experiments. For instance, the PAMPA assay detects passive transport only, while Caco-2 cells include transporters. A comparison between transport in PAMPA and Caco-2 cells by a calibration plot reveals compounds with greater or lesser transport in Caco-2 cells than in PAMPA. These compounds should be tested in uptake and efflux transport assays in order to gain a deeper insight into absorption fate.

Several *in silico* models for prediction of oral absorption are available [133–136]. Simple models are based on only few descriptors like $\log P$, $\log D$ or polar surface area (PSA), while they are only applicable if the compounds are passively absorbed. In case of absorption via active transporters or if efflux is involved, prediction of absorption is still not successful.

GastroPlus [137] and iDEA [138] are absorption-simulation models based on *in vitro* input data like solubility, Caco-2 permeability and others. They are based on advanced compartmental absorption and transit (ACAT) models in which physicochemical concepts are incorporated. Both approaches were compared, and shown to be suitable to predict the rate and extent of human absorption [139].

Problems related to poor systemic exposure are also tied to volume of distribution, which is indirectly related to plasma protein binding. Human plasma contains more than 60 different proteins of which the major components are serum albumin (HSA, 60%) and glycosylated proteins (AGP). Eighteen different variants arising from single amino acid mutations have been identified, accounting for different protein binding. Allelic variation makes data consistency difficult and hence modeling of the resultant data less reliable [140].

Prediction of bioavailability from molecular structure is quite difficult, since bioavailability depends on absorption and first-pass clearance [141]. By applying "fuzzy adaptive least squares", Yoshida and Topliss generated a QSAR model using $\log D$ at pH 7.4 and 6.5 as input for physicochemical properties and the presence/absence of certain functional groups as structural input.

They achieved a classification of drugs into one of four bioavailability categories with an overall accuracy of 60% [142].

First-pass clearance can be tracked to gut stability or metabolism by phase I and then either direct clearance or clearance of the metabolite by phase II enzymes or biliary, renal or plasma clearance. Metabolite stability by phase I enzymes includes inhibition, induction, regiospecificity, lability or affinity toward several cytochrome P450 enzymes or flavin monooxygenase (FMO). Inhibition might cause drug–drug interactions and can be related to competitive, noncompetitive, uncompetitive or mechanistic inhibition [143]. Metabolite-based inhibition of the P450 enzymes also needs to be considered if toxicity aspects are related to these enzymes. Assays for differentiating these various types of mechanisms of inhibition are necessary and need to be applied for a reasonable number of compounds in a series in order to apply molecular modeling techniques to help in designing proper molecules with the preferred inhibition properties. Typical fluorimetric assays do not differentiate substrates that can be competitively bound or those that can covalently modify and inactivate the P450 enzymes. Only by different experimental approaches (time, substrate concentration, inhibitor concentration, NADPH-dependent inhibition) can these types of compounds be identified. Sometimes, the identification of metabolites helps to decipher the binding modes in the cytochrome P450 responsible [144]. Despite these problems, some successful applications of modeling to reduce drug–drug interactions, especially for CYP2D6 and CYP2C9, have been reported (see below).

Assays for FMO and most phase II enzymes are typically not included as part of a standard eADMET profile and, therefore, modeling related to clearance by phase II enzymes has not been attempted, although structural information for some sulfotransferases is available [145].

In silico methods to predict metabolism are based on QSAR, 3-D-QSAR or protein and pharmacophore models [146]. Early predictions of metabolism within a particular compound are now feasible. One approach by Korzekwa and coworkers uses, for example, reaction energetics to develop a predictive model for CYP3A4-mediated metabolization [147]. Sheridan and coworkers [148] described a model to predict likely sites of CYP3A4-mediated metabolism based on the energy necessary to remove a hydrogen radical from each site, as estimated by AM1 semiempirical molecular orbital calculations and the surface area exposure of the hydrogen atom. The development and validation of a general quantitative structure–property model of metabolic turnover rates in human S9 homogenate based on uniform biological data of 631 diverse compounds proprietary to GlaxoSmithKline has also been described [149]. This model was able to classify 83% of test compounds correctly for their metabolic stability. Other approaches are based on databases to predict metabolism, e.g. MetabolExpert (Compudrug), META (MultiCASE), Meteor (Lhasa) and the MDL database Metabolite [150].

4.3 Prediction of Toxicity

Approximately 20–40% of drug failures are attributed to toxicity problems. Hence, *in silico* predictive toxicology tools are necessary to reduce attrition. Most of the current software packages available deal with carcinogenicity and mutagenicity [151], and are based on available public domain toxicological data. *In silico* tools for other important end-points such as hepatotoxicity, QT prolongation [152] and phospholipidosis are emerging, and are expected to improve the design and optimization of novel compounds.

4.4 Physicochemical and ADMET Property-based Design

Property-based design supplements successful activity-based strategies to produce drug candidates [153]. Property screening in parallel with activity screening allows medicinal chemists to simultaneously optimize both biological activity and drug-like properties, which is a widely accepted and implemented approach within the pharmaceutical industry. ADMET filters are now applied even at very early stages of drug design, e.g. in virtual screening. The first generation of predictive ADMET tools allows focusing on compounds with a high potential of the required pharmacokinetic and safety criteria, and should reduce compound attrition. However, all currently available models are based on a limited set of data and are therefore restricted in their applicability. Hence, a continuous improvement and refinement of these models is required by taking into account more data that is also of high quality. Thereby, ADME might be considered as "automated decision-making engine" in the early discovery paradigm and as ADME in the regulatory phase [154].

5 ADME/Antitarget Models for Lead Optimization

5.1 Global ADME Models for Intestinal Absorption and Protein Binding

This section will provide an overview of ADME models from our group to illustrate our approach to building predictive models on structurally diverse training sets. Data sets for intestinal human absorption and human serum albumin binding are discussed, while models for other relevant ADME properties have also been obtained. Those models, however, do not stand alone, but are used in combination with those models tailored for affinity and selectivity in the frame of multidimensional lead optimization.

Recently, a set of alignment-free 3-D descriptors named VolSurf was developed by Cruciani and coworkers [155], referring to molecular properties extracted from 3-D molecular interaction fields of various probe atoms. VolSurf transforms the information present in GRID-derived [113] 3-D fields for particular probes into quantitative descriptors (Figure 5), which carry information related to pharmacokinetic properties like polarity, hydrogen bonding, lipophilicity, size, polarizability and others. The descriptors are easy to understand and to interpret, thus providing guidelines for chemical optimization after a linear model has been established. In most cases, the GRID water, hydrophobic (DRY) and carbonyl oxygen probes were utilized,

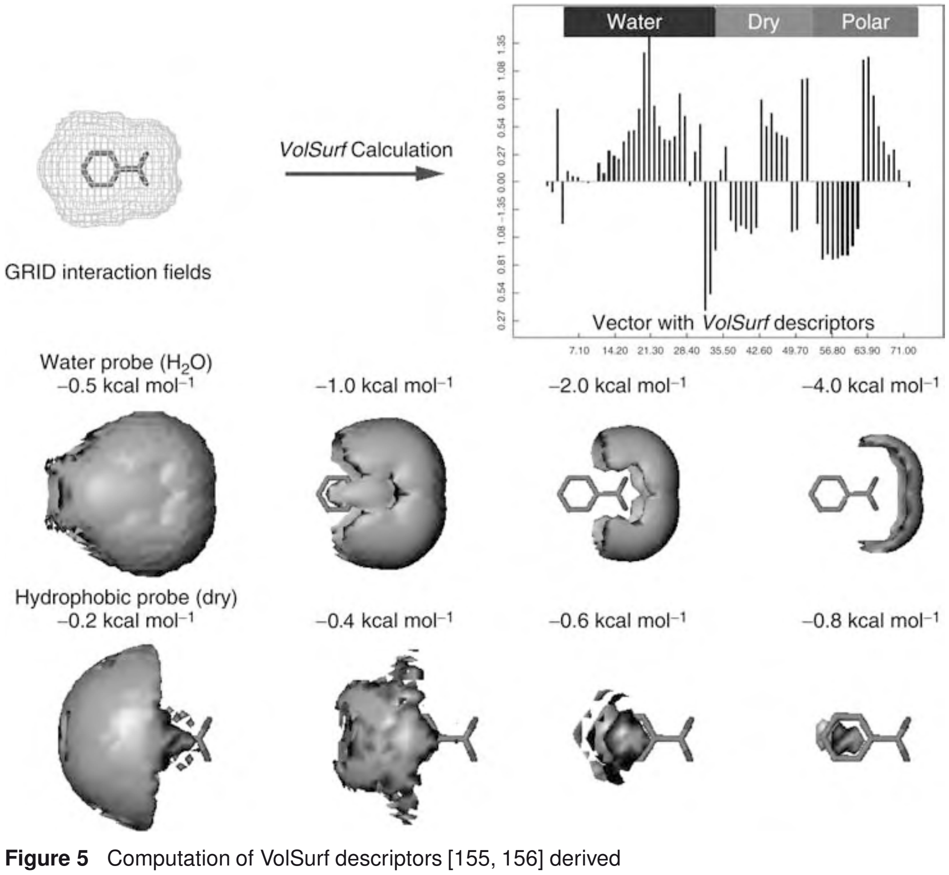


Figure 5 Computation of VolSurf descriptors [155, 156] derived from GRID molecular interaction fields. Interactions of the example molecule with a water and dry probe at different contour levels are used to compute a vector of 72 volume-, size- and surface-based descriptors.

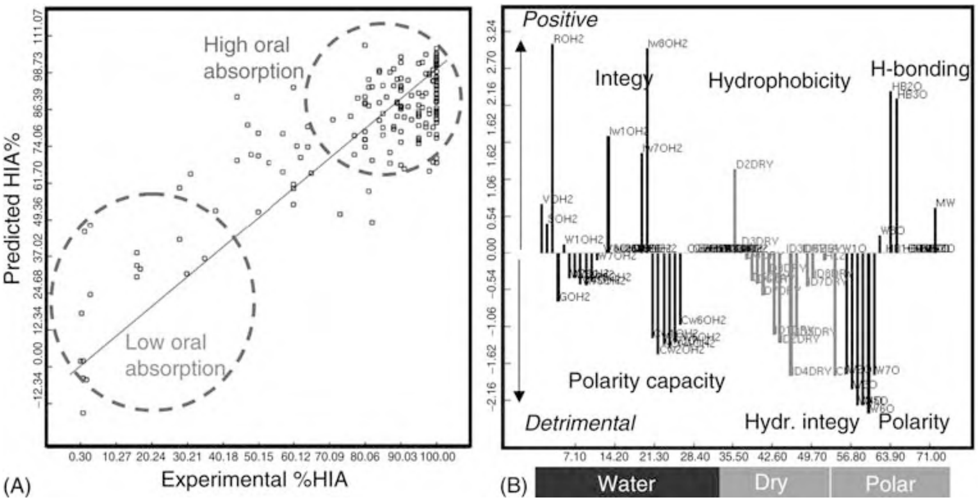


Figure 6 Correlation of VolSurf descriptors with human intestinal absorption for 169 drug molecules. (A) Predicted versus experimental percent HIA (human intestinal absorption) from the final PLS model with q^2: 0.662, r^2: 0.709 and 4 PLS components. (B) PLS loadings for a four-component model showing the importance of individual VolSurf descriptors to the prediction of human intestinal absorption.

as interaction of molecules with biological membranes is mediated by surface properties like shape, electrostatic forces, hydrogen bonding and hydrophobicity.

These descriptors have been reported in the literature to correlate with bioavailability, blood–brain partitioning, membrane transport and other properties [156–159]. They are also correlated to relevant physicochemical properties, and were successfully applied to many internal and public data. For example, we derived PLS models [160] using 72 VolSurf descriptors for HSA binding using 95 drugs on a data set from Colmenarejo and coworkers [161] ($r^2 = 0.76$, $q^2 = 0.67$, 6 PLS components) and for human intestinal absorption using data of 169 drugs compiled by Zhao and coworkers [133] ($r^2 = 0.71$, $q^2 = 0.66$, 4 PLS components). Those models were obtained using a canonical 3-D structure of neutral molecules.

As an example, the correlation of VolSurf descriptors to the human intestinal absorption for 169 drugs is shown in Figure 6. The conclusions for factors influencing permeability and absorption are in agreement with earlier findings, pointing to the positive impact of hydrophobicity, integy moment, shape and hydrogen bonding, while polarity derived using the GRID water and carbonyl O probes as well as the capacity factors from polar interactions on the entire surface are detrimental for intestinal absorption. However, because of the nature of these descriptors, this approach led to an enhanced understanding of the physicochemical requirements for a pharmacokinetic

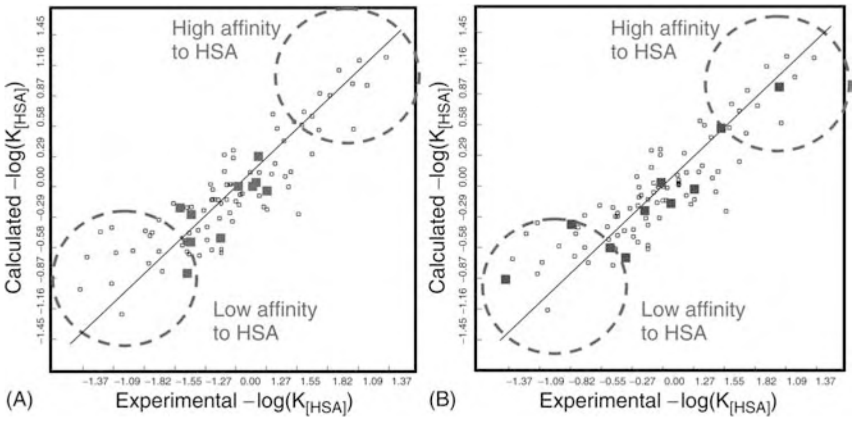


Figure 7 Correlation of VolSurf descriptors with HSA binding affinity. Two submodels indicate their predictive ability to external test sets. Ten compounds were removed by either experimental design on PCA scores (A) or literature proposals (B). Model A: q^2: 0.668, r^2: 0.763, 6 PLS components, SDEP for prediction 0.223 for 10 cpds. Model B: q^2: 0.646, r^2: 0.745, 6 PLS components, SDEP for prediction 0.274 for 10 cpd.

effect. For example, the balance between lipophilic and hydrophilic parts in combination with size, volume and other effects guides the design of new compounds.

The interpretation for HSA binding affinity is also in accord with literature findings and X-ray information on warfarin binding to the first drug binding site in HSA [162]. For instance, hydrophobicity, volume, shape and molecular weight increase HSA binding, while factors like polarity, integy moment, hydrogen bonding and favorable interactions to water are detrimental. Typically, hydrophobicity and geometrical factors like shape are reported as essential in the literature [161]. These models are always validated by external test data sets, as is schematically shown in Figure 7, where 10 compounds have been removed from the HSA training data set by two different approaches and predicted using two only slightly differing models. In both cases, the agreement between experiment and prediction was very good.

Alifrangis and coworkers [158] reported a structure–property model for membrane partitioning for 20 peptides with data from two chromatographic systems with phospholipides as the stationary phase, immobilized artificial membranes (IAM) and immobilized membrane chromatography (ILC). The relationship between these measures and three different sets of calculated descriptors (molecular surface area, MolSurf and VolSurf) were analyzed using PLS, showing that VolSurf-derived models are superior to both others [158]. In particular, the VolSurf critical packing descriptor to describe interactions of amphiphilic molecules with membranes is important to explain the membrane partitioning ability. This agrees with internal VolSurf models derived for PAMPA membrane transport [163] to understand passive transcellular transport across membranes. One of our internal models based on 29 compounds characterized by immobilized artificial membrane chromatography by Salminen and coworkers [164] shows $r^2 = 0.81$ and $q^2 = 0.70$ for two PLS components derived using VolSurf descriptors. This is one of the rare examples where ionized starting molecules led to slightly better PLS statistics, while the general chemical interpretation was not affected.

VolSurf was also successfully applied in the literature to predict absorption properties [156] from experimental drug permeability data of 55 compounds [165] in Caco-2 cells (a human intestinal epithelial cell line derived from a colorectal carcinoma) and Madin-Darby canine kidney (MDCK) cell monolayers. In this interesting case, it was shown that models including counterions for charged molecules clearly show significantly better quality and overall performance. The final model was also able to correctly predict, to a great extent, the relative ranking of molecules from another Caco-2 permeability study by Yazdanian and coworkers [166].

5.2 Selected Examples to Address ADME/Toxicology Antitargets

For ADME and toxicology properties, which are related to a particular target protein, approaches similar to those for affinity and selectivity are used in our group, while most of the ligand-based studies have recently been augmented by X-ray structures and/or validated homology models. Again, both approaches led to significant results in combination, when applied in the context of lead optimization in internal projects.

The K^+ channel encoded by the human ether-à-go-go-related gene (hERG) is one of the many ion channels that are crucial for normal action potential repolarization in cardiac myocytes [152]. This hERG channel is connected to drug-induced long QT syndrome causing cardiac toxicity of a wide range of pharmaceutical agents. This undesirable side-effect for noncardiac drugs has caused recent withdrawals of drugs from the market and stimulated many studies to establish a structural hypothesis for hERG and a structure–activity relationship of hERG channel blockers. Given the wide range of chemical structures that act as hERG inhibitors and the observation that different compounds are reported to bind to different states (open or closed) of the channel, multiple binding sites have been proposed [152]. On the basis of a homology model of the closed form of the tetrameric pore derived from the X-ray structure of the bacterial K^+ channel KcsA [167] combined with site-directed mutagenesis studies, several amino acids were identified as important for high-affinity binding of the methansulfonanilide MK-499 and thus provided a structural basis for drug-induced long QT syndrome [168]. Interestingly, two aromatic residues, Tyr652 and Phe656, lining the inner pore as a unique structural motif in hERG compared to other K^+ channels (except EAG) are proposed to interact with aromatic groups in ligands.

Perlstein and coworkers [169] describe the status of previously developed hERG structure–activity *in silico* models. While some of the models are intended to be used as early filters in compound design [170], only the 3-D-QSAR studies of Cavalli and coworkers [171] and the combination [172, 173] of 3-D-QSAR from consistent biological data with homology models of the closed and open form of hERG built from the low-resolution crystal structure of the MthK K^+ channel [174] allow us to understand structural reasons for hERG blocking, and thus provide sufficient details to apply them during the multidimensional lead optimization of a candidate molecule. These models have been internally derived from consistent biological data on literature compounds and close analogs, while they are constantly updated using novel molecules and biological data from lead optimization programs. Many internal applications have proven the value in using such a combined approach. The essential elements of the hERG pharmacophore, the 3-D-QSAR and the homology model are as follows. (i) The hydrophobic feature optimally consists of an aromatic group that is capable of engaging in π-stacking with Phe656. Optionally, a second aromatic or hydrophobic group may contact an additional Phe656 side-chain. (ii) The basic nitrogen appears to undergo a π-cation interaction with Tyr652. (iii) The pore diameter and depth of the selectivity loop relative to the intracellular opening, act as constraints on the conformation-dependent inhibitor dimensions [172]. However, undesirable structural features for hERG are not sufficient in the context of a lead optimization project. It is important to understand which variations influence hERG binding, but are still tolerated by the desired biological target. Both properties were used as lead optimization parameters by Friesen and coworkers [175] for PDE-4 inhibitors in a systematic approach to identify structural features that only affect hERG binding.

Toxicity caused by drug–drug interaction also resulted in withdrawal of several drugs from the market. Compounds that are potent inhibitors of the major metabolizing enzymes can potentially affect the metabolism of other molecules and thus lead to toxicity. Early information on cytochrome P450 inhibition thus is useful to develop structure–activity relationships and minimize the potential toxicity of development candidates. Inhibition of CYPs can be substrate dependent, which especially is reported for CYP3A4 with a very large active site that is able to bind multiple substrates in different orientations [176]. The growing knowledge about CYP substrates has increased the understanding of active-site requirements either by protein- or ligand-oriented studies [177]. However, understanding the nature of substrate specificity of each CYP requires knowledge of the interaction of drugs with each CYP active site. A detailed analysis of substrate-binding affinities and selectivities for the CYP2 family is given elsewhere [118, 178].

Pharmacophore-based models for many cytochrome P450 isoforms have been reported for CYP2B6 substrates [179], CYP3A4 substrates [180] and inhibitors [181], CYP2C9 inhibitors [182], and CYP2D6 inhibitors [183]. A 3-D-QSAR model for CYP2C9 inhibitors for prediction of drug–drug interactions was reported by Rao and coworkers [184] and compared with appropriate homology models. Afzelius and coworkers [185] used alignment-independent GRIND descriptors implemented in the program Almond [186] to obtain qualitative and quantitative predictions of CYP2C9 inhibitors for a series of structurally very diverse molecules. De Groot and coworkers reported a combined protein and pharmacophore model approach to understand and predict CYP2D6-mediated drug metabolism [187, 188]. A model for 40 different CYP2D6 substrates (hydroxylation, *O*-demethylation) was obtained by combination of pharmacophores, protein models and molecular orbital calculations. This model was extended by a second pharmacophore explaining 14 less-common *N*-dealkylation transformations. The final model was in agreement with additional substrate and site-directed mutagenesis data. Zamora and coworkers developed an interesting novel approach named MetaSite to predict the metabolization site of substrates in CYP2C9 on the basis of a combined application of alignment-independent descriptors to describe the protein binding site taken from a CYP2C9 homology model and a distance-based representation of individual substrates [189].

The recently solved X-ray structures of rabbit microsomal CYP2C5 in complex with diclofenac [190] and a sulfaphanazole derivative [191] provided additional evidence on how the molecular recognition of structurally diverse substrates takes place. Comparisons of the complex with apo CYP2C5 [119] indicates that the protein closes around the substrate and prevents open access of water from bulk solvent to heme Fe. Multiple substrate-binding models of the sulfaphenazole derivative are in agreement with the experimentally derived ligand electron density maps and the finding that the substrate is not tightly constrained in the active site. For diclofenac, a single binding mode is consistent with the observation of a highly regiospecific hydroxylation at the distal ring in 4′ position. This large active site and the striking enzyme flexibility upon binding of both ligands underlines the ability of the drug-metabolizing enzymes to work on structurally diverse substrates of different sizes.

This work was complemented by the X-ray structure of human CYP2C9 in complex with warfarin by Williams and coworkers [192] (Figure 8). This report provides the first human CYP protein structure, while the warfarin-binding mode reveals previously unanticipated interactions in a new binding

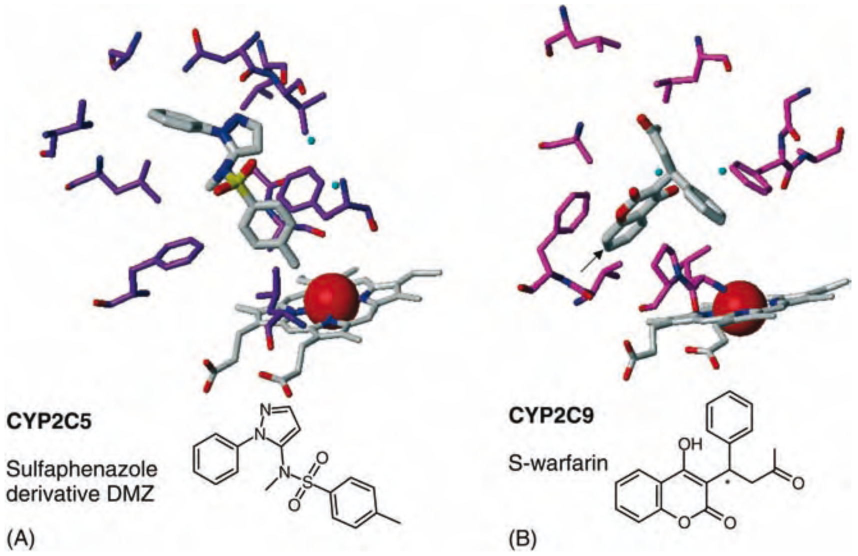


Figure 8 Experimental ligand interactions with the cytochrome P450 2C family. (A) X-ray structure of the sulfaphenazole derivate DMZ in rabbit CYP2C5 at 2.3-Å resolution (PDB 1N6B) from Wester and coworkers [191]. Only one of the two ligand orientations for DMZ in accord with electron density is shown placing the benzylic methyl group 4.4 Å from the heme Fe. (B) X-ray structure of S-warfarin in human CYP2C9 at 2.55-Å resolution (PDB 1OG5) from Williams and coworkers [192]. The substrate is situated in a predominantly hydrophobic pocket. This binding mode places the 6- and 7-hydroxylation sites 10 Å from the iron (arrow).

pocket. From the binding mode, the authors conclude that CYP2C9 may undergo an allosteric mechanism during its function and that two molecules for warfarin could be accommodated in the very large CYP active site. Collectively, these X-ray structures provide insights into relevant protein–ligand interactions for particular human CYP subfamilies, and thus can be used for docking studies and building scaffold-specific 3-D-QSAR models driven by a protein-derived alignment rule. Those models provide guidelines on where to optimize a molecule with a CYP liability. Hence, a higher-resolution picture of drug–CYP interactions begins to emerge for some subfamilies, allowing the use of this information as one optimization parameter in multidimensional optimization projects.

6 Integrated Approach

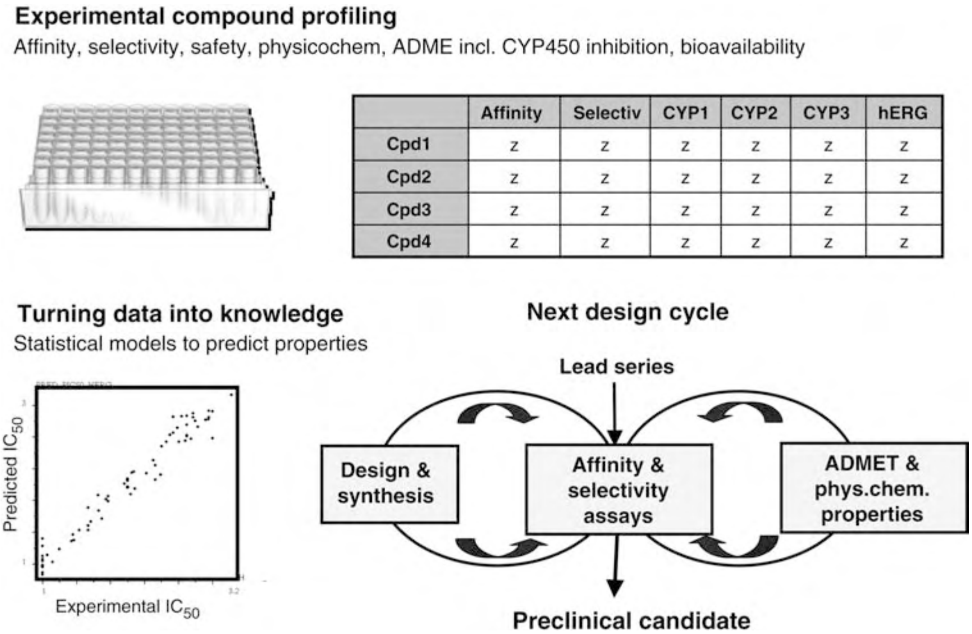
6.1 Strategy and Risk Assessment

While previous sections discussed computational approaches towards the understanding and analysis of individual properties like affinity, selectivity and ADME parameters, here we show how those tools should be combined to arrive at a lead optimization strategy that is able to manage and design for multiple properties in lead optimization cycles. While in early phases of discovery without knowledge of multiple analogs for a particular hit, "global" models for ADME properties are applied for library design and selection of promising hits, this focus is shifted toward "local" models based on information about analogs during the subsequent optimization phase. The lack of consistent publicly available data on properties like oral absorption prompted us to optimize pharmacokinetic properties based on consistent biological data for one series.

Any significant improvement of the lead optimization process requires predictive *in vitro* assays, models and computations for ADME and toxicological properties to be incorporated into the iterative cycle of compound design and synthesis. In this manner, compounds can be optimized in parallel by a multidimensional optimization strategy, which integrates the evaluation and optimization for affinity, selectivity, physicochemical and ADME properties. This approach, however, requires a proper experimental design of compounds related to a specific chemotype prior to any synthesis. Such a design should take into account a broad coverage of the chemotype-specific chemical space. The reliability of any subsequent statistical model strongly depends on the information content of the training set data. If experiments are planned inefficiently, resulting in higher numbers of experiments, less information and sometimes misleading data are generated. The systematic variation of one factor at a time is only appropriate if the response surface is not influenced by interactions between factors, while cooperative effects may require advanced design approaches [193]. Thereby, appropriate chemotype-specific local models can be built and subsequently applied in the design of new compounds.

Needless to say that reliable experimental data based on a single mechanism should be preferred. Such single-mechanism data provide not only a deeper understanding of the underlying molecular mechanisms, but also allow us to derive better, less noisy and thus more predictive *in silico* models with information on directions to be taken for further design. Furthermore, this approach allows early identification of properties to be improved in lead optimization and, in addition, whether two, three or more variables can be optimized in parallel. For example, if the optimization of the biological properties $Y1$ and $Y2$ requires increasing hydrophobicity for better activity, while this descriptor is detrimental for the improvement of a third property $Y3$, a detailed statistical assessment is necessary whether it is possible at the end to optimize all three dependent variables in parallel to derive a compound with a suitable target product profile.

In particular, parallel optimization of affinity/selectivity and pharmacokinetic properties are difficult to achieve, especially if several pharmacokinetic parameters need to be optimized (e.g. absorption and volume of distribution). The number of variables to be optimized as well as their optimization potential should always be assessed in order to deal with a minimal set of properties to work with. It is our experience in multiple lead optimization projects that as more variables need to be addressed in parallel, the more difficult, time consuming and exhaustive a particular lead optimization will be. In general, the probability of a successful lead optimization of a certain chemotype resulting in a development candidate decreases substantially with


Experimental compound profiling
Affinity, selectivity, safety, physicochem, ADME incl. CYP450 inhibition, bioavailability
Turning data into knowledge
Statistical models to predict properties
Predicted IC50
Experimental IC50
Next design cycle
Lead series
Design & synthesis
Affinity & selectivity assays
ADMET & phys.chem. properties
Preclinical candidate

	Affinity	Selectiv	CYP1	CYP2	CYP3	hERG
Cpd1	z	z	z	z	z	z
Cpd2	z	z	z	z	z	z
Cpd3	z	z	z	z	z	z
Cpd4	z	z	z	z	z	z

Figure 9 Individual components of multidimensional optimization. This approach requires experimental compound profiling against key properties, which should be performed on a designed compound subset to maximize information with a minimum number of molecules. These data are used to derive models for key properties, which are applied during the next design cycle. The results then led to augmented models. The process is characterized by a tight integration of *in vitro* and *in silico* tools for profiling compound series to guide chemical optimization.

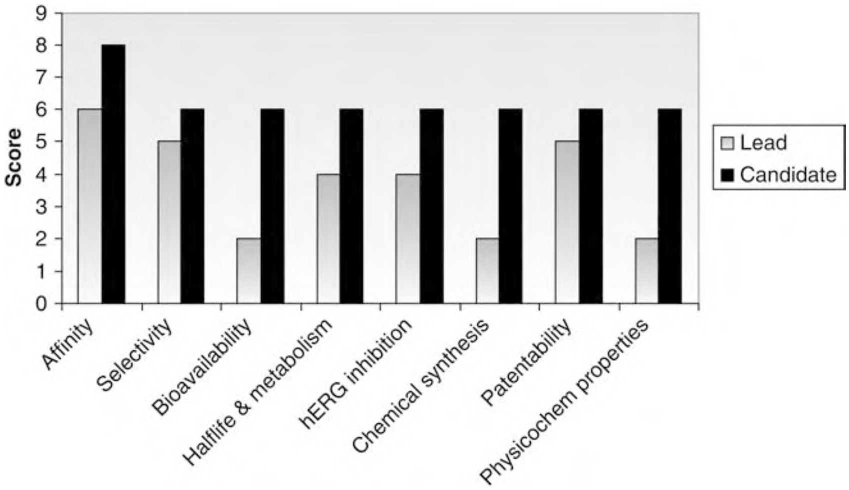


Figure 10 Example for multidimensional optimization on relevant properties during the lead optimization phase from leads to a development candidate. After some iteration, compound properties are either improved or show no further optimization potential.

the need to improve more than three properties in order to fulfill the target property profile.

This requires risk estimation by an early assessment of the optimization potential for any novel chemistry series in order to obtain early information on ADME and antitarget properties as well as potential toxicity issues, which are known to become increasingly important for drug failures. Unfortunately, only a few ADME antitargets have been addressed already (e.g. hERG activity, see above), while others need to be improved. Toxicity, however, is quite often not related to a single mechanism and therefore very difficult to address [194]. Certainly, more reliable and more predictive SARs for target affinity as well as for antitarget and pharmacokinetic properties will strengthen the multidimensional optimization of compounds. The integration of these tools into connected processes will certainly raise modern drug discovery to a new level (Figure 9).

6.2 Integration

The ideal case during lead optimization is to use a consistent set of descriptors for multiple biological variables. This concept is followed wherever applicable, although this is not always possible from a practical point of view. If such a single set of informative descriptors, e.g. VolSurf, to model binding affinity in structure- or ligand-based design and ADMET properties at the same time [21] is used to build appropriate models for all dependent

(Y) variables, an early assessment of the optimization potential is feasible. The approach described by Zamora and coworkers [21] is the promising integration of both binding affinity and passive pharmacokinetic properties on the descriptor level.

However, if different models and descriptors are used and applied, certain criteria, decision trees or scoring functions need to be used to deal with the multidimensional optimization. Decision trees refer to approaches similar in spirit to the well-established "Rule of Five" and are often project-specific variations and improvements thereof. The chemical meaning and interpretability of the descriptors is essential here. Scoring models, as a slightly different approach, condense each *in silico* estimated compound property into a range between 0 and 1 based on project team-specific thresholds. Some degree of uncertainty could be considered by introducing a linear increase in this score contribution from 0 (undesirable) to 1 (desirable) within a narrow area around the threshold. Those individual scores are compiled into a weighted sum and used to evaluate novel synthesis proposals, often biased by the knowledge of the protein 3-D structure (Figure 10). Step-by-step application of decision trees or a set of single models, however, might ultimately sort out almost all compounds depending on the sequence used and are therefore less valuable. Scoring functions, however, offer the opportunity to fine-tune the function in terms of the most important variables without neglecting any of the other properties. The thoughtful integration of *in silico* tools in this multidimensional optimization process will certainly improve candidate quality in the next decade.

6.3 Literature and Aventis Examples on Aspects of Multidimensional Optimization

Although the optimization of promising lead compounds toward clinical candidates is one of the essential skills within the pharmaceutical industry, there are not many reports on the entire multidimensional optimization strategy or selected aspects. Here, we discuss examples from the medicinal chemistry literature and projects at Aventis in which either the chosen optimization strategy or the computational tools are interesting and agree with the proposed multidimensional optimization strategy and its requirements. Important prerequisites for lead optimization that are essential to successfully manage a drug discovery program include the early assessment of the optimization potential for a lead series, the risk assessment of how many parameters have to be optimized in a multidimensional context while all the others should remain in the positive range, where they have been from the beginning. The literature reports address one or only a few additional parameters in addition to biological affinity at the target enzyme to progress a series. This could be seen as typical examples for one single optimization cycle within multidimensional lead optimization.

Linusson and coworkers at AstraZeneca described the application of statistical molecular design for planning and analysis of a parallel synthesis lead optimization library of thrombin inhibitors [195]. This structurally well-characterized serine protease is involved in the blood coagulation cascade and was targeted by many industrial drug discovery projects. The report described how building blocks were selected for a central scaffold with three vectors directed toward individual thrombin subpockets S1, S2 and S4. This was done on the basis of a quantitative statistical analysis of known experimental information such as biological affinities for related thrombin inhibitors at AstraZeneca. The focus of the planned library was to replace benzamidine, which provides important polar interactions within the protease S1 pocket, while it has been recognized for a while that benzamidine substituents are detrimental for oral bioavailability, mainly due to their low intestinal membrane permeability, which could be attributed to their high pK_a value. Hence, parameters like pK_a and membrane permeability are seen as equally important in the planning phase for multidimensional optimization. There are many reports on benzamidine-based thrombin inhibitors, which are all end-points in lead optimization, and lead to potent inhibitors in terms of affinity, selectivity and anticoagulation effect, while significant oral bioavailability could not be demonstrated, except for prodrugs like ximelagatran [196]. Considering this property for a development candidate very late in the discovery phase, however, often results in a significant loss of affinity when replacing the S1 benzamidine. These considerations led the authors to include topologically similar S1 directed substituents with a potential for lower pK_a into their statistical selection procedure of appropriate building blocks. The resulting parallel synthesis library was analyzed with respect to affinity, membrane permeability, pK_a and trypsin selectivity. To this end, the authors derived multivariate QSAR models for all key biological data (affinity, selectivity, pK_a, permeability). As the library has been carefully designed, the final models, although only based on a limited series of molecules, are significant and predictive, and might guide the direction for further optimization. The SAR information for key properties, which was only found using statistical molecular design in combination with multivariate analysis, could now be applied to focus a second follow-up library. This interesting study combines the concepts of multidimensional optimization and suggests statistical approaches to obtain informative models. As this study presents only one cycle in lead optimization, it is only a "snapshot" in the search for a development candidate.

Sugano and coworkers studied the membrane permeation of 51 benzamidine-based thrombin inhibitors in a rat everted sac permeability model [197]. They reported significant membrane permeabilities in this *in vitro* model, which they attributed to passive paracellular transport – a different absorption mechanism to transcellular permeability. On the basis of their evaluation and our internal predictive VolSurf model [160] for this series (r^2: 0.81, q^2: 0.60, 4 PLS components), it can be concluded that factors like size and shape, which had previously been reported to affect paracellular permeability, are indeed important in the VolSurf PLS model to explain the local structure–permeability relationship of one particular scaffold. Hence, local statistical models provide a qualitative ranking of candidates and thus are valuable for optimization of pharmaceutically relevant compounds, especially if combined with additional models to understand affinity, selectivity or any particular pharmacokinetic behavior.

Burgey and coworkers at Merck described their approach toward metabolism-directed optimization of 3-aminopyrazinone-based thrombin inhibitors [198] to result in an orally bioavailable series. Several research groups have now successfully replaced the benzamidine moiety in S1 by less-basic or non-basic substituents with different protein–ligand interactions to overcome the specific benzamidine problems described above, resulting in series with increased oral bioavailability and pharmacokinetics. Starting from an aminopyrazinone–acetamide scaffold with moderately basic 2-aminopyridine as an S1-directed benzamidine mimetic (Figure 11), they discovered three main sites of metabolism, which they related to the observed insufficient pharmacokinetic behavior. The optimization was dictated by metabolic considerations in concert with required target affinity. On the basis of the available experimental information on binding modes for this chemotype and the detailed drug metabolism and pharmacokinetic studies to unravel the mechanistic origin of metabolic instability, they were able to design a series of metabolically more stable variations of the original lead structure with similar enzyme affinity and selectivity. The introduction of metabolically more stable substituents at this scaffold led to a final compound with improved pharmacokinetic properties. Two key observations are essential in this and other successful studies: (i) the careful balance of decreasing thrombin affinity by metabolically more stable substitutions in one subpocket by compensation within another pocket, which could have been achieved only using the wealth of structural knowledge and understanding of thrombin protein–ligand interactions, and (ii) the careful correlation of *in vivo* pharmacokinetic observations to their *in vitro* origin and the subsequent monitoring of each metabolically less labile substituent in appropriate *in vivo* dog pharmacokinetic studies.

In a second contribution from the same group, a subsequent optimization cycle for the previous scaffold was directed toward improving solubility while reducing the number of chemical steps in the overall synthesis of S1 replacements. The readily available S1 directed building blocks then led to

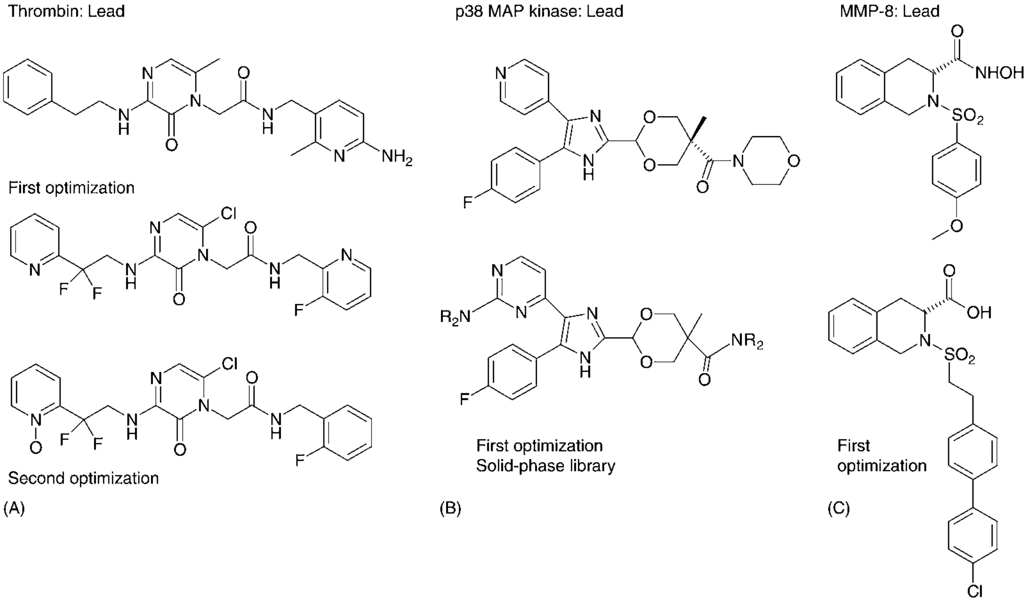


Figure 11 Selected examples for lead optimization under consideration of multiple parameters simultaneously: (A) thrombin inhibitors, (B) p38 mitogen-activated protein (MAP) kinase inhibitors and (C) MMP-8 inhibitors.

establish the structure–activity relationships of three key parameters besides thrombin affinity, i.e. oral bioavailability, half-life and human liver microsome stability. Hence, the optimization strategy of the S1 subpocket was again guided by pharmacokinetic considerations, which were related by *in vitro* solubility assays. Finally, the introduction of S4 pyridine *N*-oxides with increased solubility led to a new orally bioavailable series. An expedited investigation of the P1 SAR incorporating several S1 *N*-benzylamides with respect to oral bioavailability, plasma half-life and human liver microsome stability subsequently revealed an interesting candidate for advanced pharmacological evaluation after careful monitoring and optimization of a variety of affinity, selectivity, physicochemical and pharmacokinetic parameters.

Hasegawa and coworkers from Roche [199] described a local model to understand structure–pharmacokinetic relationships for 107 benzofurans as *N*-myristoyltransferase inhibitors, based on cassette-dosing pharmacokinetic studies and rat elimination half-lifes as dependent variables. They obtained a relatively simple, yet effective, statistical model, which they describe as useful and which gives a direction for designing new inhibitors having good pharmacokinetic profiles. Similar local QSAR models for relevant pharmacokinetic properties could be used in combination with affinity prediction models or docking and scoring calculations, if applicable, to select novel synthesis candidates from a series of feasible proposals in a multidimensional fashion.

McKenna and coworkers at Aventis described the design and synthesis of a solid-phase library to optimize a pyridine-imidazole-based p38 MAP kinase inhibitor toward target affinity and rat oral bioavailability [200]. The authors describe a computational approach toward the design of a library with 570 analogs on two attachment points. Those are selected using a Monte Carlo monomer selection (MCMS) strategy on the basis of combinatorial synthetic efficiency on one hand, while maximizing the estimated bioavailability, as considered by *in silico* descriptors like PSA and a modified "Rule of Five" definition. MCMS applies a scoring function controlled by a series of weights for individual components to drive monomer selection toward solutions that satisfy the design constraints [201]. From this focused lead-optimization library, 10 relatively structurally diverse compounds with improved potency and acceptable bioavailability and pharmacokinetic parameters were rapidly identified as follow-up to the previous candidate.

In their search for nonbenzamidine-based inhibitors of the serine protease factor Xa, Choi-Sledeski and coworkers [202] described the discovery of an orally efficacious inhibitor that incorporates a neutral substituent directed toward the protease S1 pocket as a result of an optimization strategy guided by structure-based rational design and a qualitative consideration of bioavailability. The conflicting and nonconclusive SARs of initial derivatives on that scaffold could only be resolved by X-ray structure analysis of a relevant protein–ligand complex [64]. The X-ray structure revealed a strong preference for a neutral substituent directed toward this S1 pocket for the explored ketopiperazine scaffold, which offered a new perspective on designing nonbasic novel factor Xa inhibitors with increased potential for oral bioavailability. This striking preference for neutral S1 substituents was earlier reported for bioavailable thrombin inhibitors as well [203]. By minimizing the size and lipophilicity of the S4-directed substituents and by incorporating hydrogen-bonding groups on the N-terminus or on the 2-position of the S1-directed ring system, a series of active thrombin inhibitors with good bioavailability profiles was reported that showed a unique binding mode after X-ray structure analysis in thrombin. Both optimizations on serine protease inhibitors were supported by structure-based design techniques, while the latter, including favorable protein–ligand interactions, alone is not sufficient to arrive at bioavailable inhibitors. Here, the careful analysis of available X-ray structures helps to identify substituent positions that will not affect enzyme activity, but will modulate other properties favorably. Replacing a high-affinity substituent requires, on the other hand, a careful monitoring of the loss of affinity, possible substituents in other areas that might compensate for this loss and the favorable change in the desired ADME or pharmacokinetic property.

A similar situation was experienced in the search for potent and bioavailable inhibitors of the matrix metalloproteinase MMP-8 [99, 204]. For the explored tetrahydroisoquinoline scaffold, hydroxamic acids for zinc binding in 3-position are essential for matrix metalloproteinase affinity in early inhibitors, while those showed insufficient pharmacokinetic properties and low oral bioavailability. Driven by X-ray and 3-D-QSAR, alternative zinc-binding groups like carboxylates were investigated, while the expected loss in binding affinity could be compensated by optimally filling the proteinase S1′ pocket close to the catalytic zinc. The design and SAR for this series is in good agreement with those protein requirements and, moreover, monitored multiple properties including selectivity against the undesirable MMP-1 [204]. For several MMP-8 inhibitors, the oral bioavailability from rabbit *per os* studies could be correlated again to VolSurf descriptors (r^2: 0.65, q^2: 0.42, 4 PLS components), which led to a semiquantitative model [160], used in conjunction with structure-based docking and scoring, 3-D-QSAR-based affinity and selectivity predictions, and *in silico* ADME models to estimate membrane permeability, solubility and other key properties for the optimization process in this series (Figure 12). Hence, in this as well as in other series, multiple models are collectively utilized to rank and prioritize novel synthesis candidates, and focus virtual libraries.

In the search for bile-acid resorption inhibitors (BARI), a predictive 3-D-QSAR pharmacophore model for the ileal Na^+/bile acid cotransporter was derived, which enhanced the understanding of binding and transport properties [205]. This model was then also successfully explored to search for potential substitution sites, which are not relevant for the SAR of this series, while they allow the addition of additional substituents to minimize the oral uptake of inhibitors.

The approach discussed to use VolSurf derived *in silico* models to understand structure–pharmacokinetic relationships for pharmacokinetic properties was also applied to one series of selective cardiac KATP channel blockers [160]. It was found that compounds fulfilling the predefined selectivity profile exhibit only less-optimal pharmacokinetic properties because of a short plasma mean residential time (MRT). Consequently, the MRT for 28 compounds from rabbit intravenous studies for one series was used as the dependent variable to derive a VolSurf PLS model in addition to ligand-affinity SAR data. The chemical interpretation of the VolSurf model ($r^2 = 0.89$, $q^2 = 0.76$, 3 PLS components) reveals hydrophobic interactions and the hydrophobic integy moment to be strongly correlated to MRT. This model was then used to prioritize a novel set of promising synthesis candidates from a virtual library of potential products accessible via parallel synthesis.

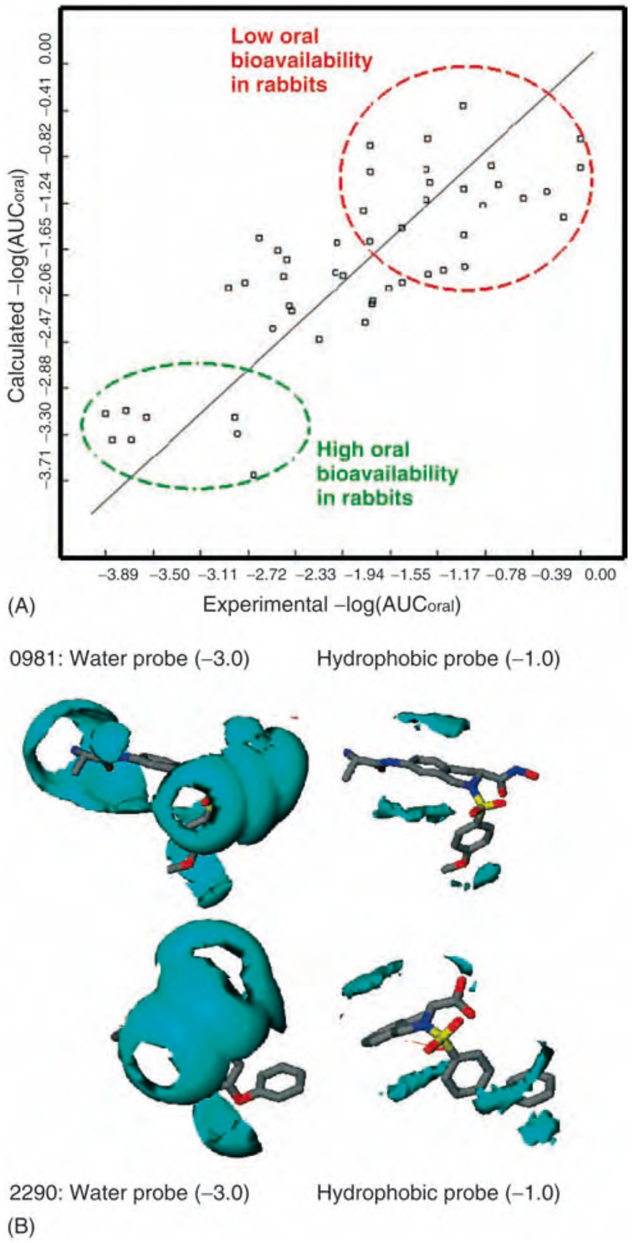


Figure 12 VolSurf model to correlate 49 matrix metalloproteinase inhibitors with different zinc-binding functionalities to rabbit oral bioavailability for metabolically stable compounds. (A) Semiquantitative PLS model (q^2: 0.424, r^2: 0.646, 4 PLS components) to rank novel synthesis candidates. Main factors influencing absorption, i.e. lower polarity, capacity factors and increased hydrophobicity, are in agreement with global models for human intestinal absorption. (B) Distribution of polar and hydrophobic surfaces for two molecules with low (0981) and higher (2290) rabbit AUC from oral pharmacokinetic studies.

Those examples illustrate typical workflows, i.e. the use of a few predictive models in parallel to rank order very focused synthesis proposals that have been derived on the basis of chemical feasibility and ligand SAR data. Each iteration and biological testing of a well-defined, information-rich set of compounds enriches our knowledge about the problem and is, of course, used to update the statistical models in order to guide the next optimization cycle. On the other hand, it might also happen that models for different properties reveal that key parameters influence different properties in opposite directions. This indicates that optimization might be extremely difficult, if not impossible, because of the very narrow balance and possibilities in the design of novel analogs.

7 Conclusions

Optimization of early lead compounds into promising drug candidates for pharmaceutical development is one of the key technologies in today's drug discovery efforts. Several medicinal chemistry approaches have been successfully applied to this problem, while the flexible integration of approaches to form an adaptive project team strategy is essential. The high complexity of this task prompts for an early identification of critical success factors and a risk assessment before individual series are progressed. High throughput alone, in this phase, does not provide a solution, as only the scientific understanding of critical success factors will enable a project team to focus on the most promising series. Therefore, a clear understanding of factors controlling affinity, selectivity, ADME and physicochemical compound properties of lead structures is essential to direct the medicinal chemistry strategies.

This chapter discussed approaches to simultaneously consider multiple aspects in optimization with the challenging goal to improve the efficiency of the drug discovery process. Some literature case studies on lead optimization illustrate the value of tight integration of parameters by means of simultaneous optimization. Data integration is mandatory for reusing the knowledge in subsequent design cycles toward an enhanced lead optimization and drug candidate selection process. Typical problems are, in particular, the improvement of affinity at the desired target, sufficient selectivity against closely related proteins, and ADME and toxicology antitargets (e.g. hERG). Furthermore, a parallel optimization of compound properties toward favorable physicochemical and related ADME/pharmacokinetic properties is essential. In the end, it is important to realize that no single approach will solve all problems for a series, but the challenge lies in the effective integration of these concepts depending on the individual problem. Only flexible and data-driven integration of those tools will enable drug discovery project teams to process interesting lead series and cancel less promising ones early and less cost-intensively.

The paradigm shift from critical activities from later drug development to earlier discovery phases some years ago has effectively led to a change in lead optimization and added a new dimension of complexity, while it is envisioned that from a multidimensional, data-driven process more suitable candidates in accord with the therapeutic target product profiles may emerge for the treatment of currently unmet medical needs.

Acknowledgments

The authors would like to thank Thorsten Naumann, Roy Vaz (Aventis Pharma), Gabriele Cruciani (University of Perugia) and Gerhard Klebe (University of Marburg) for stimulating discussions on several aspects of this chapter.

References

1 Lloyd, A. Modifying the drug discovery/drug development paradigm. Drug Discov. Today **1997**, 2: 397–8.

2 Wess, G., M. Urmann and B. Sickenberger. Medicinal chemistry: challenges and opportunities. Angew. Chem. Int. Ed. Engl. **2001**, 40: 3341–50.

3 Wess, G. How to escape the bottleneck of medicinal chemistry. Drug Discov. Today **2002**, 7: 533–5.

4 Lawrence, R.N. Sir Richard Sykes contemplates the future of the pharma industry. Drug Discov. Today **2002**, 7: 645–8.

5 Shah, N.D., Vermeulen, L.C., Santell, J.P., Hunkler, R.J. and Hontz, K. Projecting future drug expenditures – 2002. Am. J. Health Syst. Pharm. **2002**, 59: 131–42.

6 Dimasi, J.A. Risks in new drug development: approval success rates for investigational drugs. Clin. Pharmacol. Ther. **2001**, 69: 297–307.

7 Prentis, R.A., Lis, Y. and Walker, S.R. Pharmaceutical innovation by the seven UK-owned pharmaceutical companies (1964–1985). Br. J. Clin. Pharmacol. **1988**, 25: 387–96.

8 Kennedy, T. Managing the drug discovery/development interface. Drug Discov. Today **1997**, 2: 436–44.

9 Drews, J. Drug discovery: a historical perspective. Science **2000**, 287: 1960–4.

10 Hillisch, A. and Hilgenfeld, R. (eds). *Modern Methods of Drug Discovery*. Birkhäuser, Basel, **2003**.

11 (a) Gordon, E.M. and Kerwin, J.F. (eds). *Combinatorial Chemistry and Molecular Diversity in Drug Discovery*. Wiley, New York, USA, **1998**; (b) Jung, G. (ed.). *Combinatorial Chemistry*. Wiley-VCH, Weinheim, **1999**.

12 (a) Gallop, M.A., Barrett, R.W., Dower, W.J., Fodo, S.P.A. and Gordon, E.M. Applications of combinatorial technologies to drug discovery. 1. Background and peptide combinatorial libraries. J. Med. Chem. **1994**, 37: 1233–1251; (b) Gordon, E.M., Barrett, R.W., Dower, W., Fodo, S.P.A. and Gallop, M.A. Applications of combinatorial technologies to drug discovery. 2. Combinatorial organic synthesis, library screening strategies, and future directions. J. Med. Chem. **1994**, 37: 1385–99; (c) Madden, D., Krchnak, V. and Lebl, M. Synthetic combinatorial

libraries: views on techniques and their applications. Perspec. Drug Discov. Des. **1995**, 2: 269–85; (d) ELLMAN, J. A. Design, synthesis and evaluation of small-molecule libraries. Acc. Chem. Res. **1996**, 29: 132–43; (e) GORDON, E.M., GALLOP, M.A. AND PATEL, D.V. Strategy and tactics in combinatorial organic synthesis. Application to drug discovery. Acc. Chem. Res. **1996**, 29: 144–54; (f) FRUCHTEL, J.S. AND JUNG, G. Organic chemistry on solid supports. Angew. Chem., Int. Ed. Engl. **1996**, 35: 17–42; (g) THOMPSON, L.A. AND ELLMAN, J.A. Synthesis and applications of small molecule libraries. Chem. Rev. **1996**, 96: 555–600; (h) CHOONG, I.C. AND ELLMAN, J. Solid-phase synthesis: applications to combinatorial libraries. Annu. Rep. Med. Chem. **1996**, 31: 309–18; (i) BALKENHOHL, F., VON DEM BUSSCHE-HÜNNEFELD, C., LANSKY, A. AND ZECHEL, C. Combinatorial synthesis of small organic molecules. Angew. Chem., Int. Ed. Engl. **1996**, 35: 2288–337; (j) NEFZI, A., OSTRESH, J.M. AND HOUGHTEN, R.A. The current status of heterocyclic combinatorial libraries. Chem. Rev. **1997**, 97: 449–72; (k) BRISTOL, J.A. (ed.). Applications of solid-supported organic synthesis in combinatorial chemistry. Tetrahedron **1997**, 53: 6573–706; (l) GEYSEN, H.M., SCHOENEN, F., WAGNER, D. AND WAGNER, R. Combinatorial compound libraries for drug discovery: an ongoing challenge. Nat. Rev. Drug Discov. **2003**, 2: 222–30.

13 LANDRO, J.A., TAYLOR, I.C.A., STIRTAN, W.G., OSTERMAN, D.G., KRISTIE, J., HUNNICUTT, E.J., RAE, P.M.M. AND SWEETNAM, P.M. HTS in the new millennium: the role of pharmacology and flexibility. J. Pharm. Toxicol. Methods **2000**, 44: 273–89.

14 LAHANA, R. How many leads from HTS? Drug Discov. Today **1999**, 4: 447.

15 ZALL, M. The pricing puzzle. Mod. Drug Discov. **2001**, 4: 36–8.

16 ALANINE, A., NETTEKOVEN, M., ROBERTS, E. AND THOMA, A.W. Lead generation – enhancing the success of drug discovery by investing in the hit to lead process. Comb. Chem. High Throughput Screen. **2003**, 6: 51–66.

17 HILL, S. Biologically relevant chemistry. Drug Discov. World Spring **2001**, 19–25.

18 BRENNAN, M.B. Filtering out failures early in the game. Chem. Eng. News **2000**, **78(23)**: 63–73.

19 PROUDFOOT, J.R. Drugs, leads, and drug-likeness: an analysis of some recently launched drugs. Bioorg. Med. Chem. Lett. **2002**, 12: 1647–50.

20 OPREA, T. Virtual screening in lead discovery: a viewpoint. Molecules **2002**, 7: 51–62.

21 ZAMORA, I., OPREA, T., CRUCIANI, G., PASTOR, M. AND UNGELL, A.-L. Surface descriptors for protein–ligand affinity prediction. J. Med. Chem. **2003**, 46: 25–33.

22 BLEICHER, K.H., BÖHM, H.J., MÜLLER, K. AND ALANINE, A.I. Hit and lead generation: beyond high throughput screening. Nat. Rev. Drug Discov. **2003**, 369–78.

23 HODGSON, J. ADMET – turning chemicals into drugs. Nat. Biotechnol. **2001**, 19: 722–6.

24 HOPKINS, A.L. AND GROOM, C.R. The druggable genome. Nat. Rev. Drug Discov. **2002**, 1: 727–30.

25 LIPINSKI, C.A., LOMBARDO, F., DOMINY, B.W. AND FEENEY, P.J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Adv. Drug Deliv. Rev. **1997**, 23: 3–25.

26 VEBER, D.F., JOHNSON, S.R., CHENG, H.Y., SMITH, B.R., WARD, K.W. AND KOPPLE, K.D. Molecular properties that influence the oral bioavailability of drug candidates. J. Med. Chem. **2002**, 12: 2615–23.

27 SUNDBERG, S.A. High-throughput and ultra-high-throughput screening: solution- and cell-based approaches. Curr. Opin. Biotechnol. **2000**, 11: 47–53.

28 THIERICKE, R. High-throughput screening technologies. In HILLISCH, A. AND HILGENFELD, R. (eds.) *Modern Methods of Drug Discovery*, Birkhäuser, Basel, **2003**, 71–85.

29 SHUKER, S.B., HAJDUK, P.J., MEADOWS, R.P. AND FESIK, S.W. Discovering high-affinity ligands for proteins: SAR by NMR. Science **1996**, 274: 1531–4.

30 Hajduk, P.J., Boyd, S., Nettesheim, et al. Identification of novel inhibitors of urokinase via NMR-based screening. J. Med. Chem. **2000**, 43: 3862–6.

31 Fejzo, J., Lepre, C.A., Peng, J.W., Bemis, G.W., Ajay, Murcko, M.A. and Moore, J.M. The SHAPES strategy: an NMR-based approach for lead generation in drug discovery. Chem. Biol. **1999**, 6: 755–69.

32 Kumar, S. and Gunnarsson, K., Small molecule drug screening based on surface plasmon resonance. In Harvey, A.L. (ed.) *Advantages of Drug Discovery Technology*, Wiley, Chichester: **1998**, 97–114.

33 Nienaber, V.L., Richardson, P.L., Klighofer, V., Bouska, J.J., Giranda, V.L. and Greer, J. Discovering novel ligands for macromolecules using X-ray crystallographic screening. Nat. Biotechnol. **2000**, 18: 1105–7.

34 Lesuisse, D., Lange, G., Deprez, et al. SAR and X-ray – a new approach combining fragment-based screening and rational drug design: application to the discovery of nanomolar inhibitors of src SH2. J. Med. Chem. **2002**, 45: 2379–87.

35 Stahl, M., Todorov, N.P., James, T., Mauser, H., Böhm, H.-J. and Dean, P.M. A validation study on the practical use of automated de novo design. J. Comput.-Aided Mol. Des. **2002**, 16: 459–78.

36 Böhm, H.-J. and Schneider, G. (eds.) *Virtual Screening for Bioactive Molecules*. Wiley-VCH, Weinheim: **2000**.

37 Bajorath, J. Integration of virtual and high throughput screening. Nat. Rev. Drug Discov. **2002**, 1: 882–94.

38 Smith, A. Screening for drug discovery: the leading question. Nature **2002**, 418: 453–9.

39 Oprea, T.I., Davis, A.M., Teague, S.J. and Leeson, P.D. Is there a difference between leads and drugs? a historical perspective. J. Chem. Inf. Comput. Sci. **2001**, 41: 1308–15.

40 Teague, S.J., Davis, A.M., Leeson, P.D. and Oprea, T. The design of leadlike combinatorial libraries. Angew. Chem. Int. Ed. **1999**, 38: 3743–8.

41 Oprea, T.I. Current trends in lead discovery: are we looking for the appropriate properties? J. Comput. Aided. Mol. Des. **2002**, 16: 325–34.

42 Hann, M.M., Leach, A.R. and Harper, G. Molecular complexity and its impact on the probability of finding leads for drug discovery. J. Chem. Inf. Comput. Sci. **2001**, 41: 856–64.

43 Murray, C.W. and Verdonk, M.L. The consequences of translational and rotational entropy lost by small molecules on binding to proteins. J. Comput. Aided Mol. Des. **2002**, 16: 741–753.

44 Rishton, G.M. Nonleadlikeness and leadlikeness in biochemical screening. Drug Discov. Today **2003**, 8: 86–96.

45 Blundell, T.L., Jhoti, H., and Abell, C. High-throughput crystallography for lead discovery in drug design. Nat. Rev. Drug Discov. **2002**, 1: 45–54.

46 Boehm, H.J., Boehringer, M., Bur, D., et al. Novel inhibitors of DNA gyrase: 3D structure based biased needle screening, hit validation by biophysical methods, 3D guided optimization. A promising alternative to random screening. J. Med. Chem. **2000**, 43: 2664–74.

47 Pickett, S.D., Sherborne, B.S., Wilkinson, T., et al. Discovery of low molecular weight inhibitors of IMPDH via virtual needle screening. Bioorg. Med. Chem. Lett. **2003**, 13: 1691–4.

48 Lipinski, C.A. Drug-like properties and the causes of poor solubility and poor permeability. J. Pharm. Toxicol. Methods **2000**, 44: 235–49.

49 Greene, J., Kahn, S., Savoj, H., Sprague, P. and Teig, S. Chemical function queries for 3D database search. J. Chem. Inf. Comput. Sci. **1994**, 34: 1297–1308.

50 Sprague, P.W. Automated chemical hypothesis generation and database searching with CATALYST. Perspect. Drug. Discov. Des. **1995**, 3: 1–20.

51 Kubinyi, H. (ed.) *3D-QSAR in Drug Design. Theory, Methods and Applications*. ESCOM, Leiden: **1993**.

52 Kubinyi, H., Folkers, G. and Martin, Y.C. (eds.) *3D-QSAR in Drug Design, Vol. 2*. ESCOM, Dordrecht: **1998**.

53 Hansch, C. and Leo, A. (eds.) *Exploring QSAR: Fundamentals and Applications in Chemistry and Biology*. American Chemical Society, Washington, DC: **1995**.

54 Debnath, A.K. Quantitative structure–activity relationship (QSAR) paradigm – Hansch era to new millennium. Mini Rev. Med. Chem. **2001**, 1: 187–95.

55 Kellogg, G.E. and Semus, S.F. *3D QSAR in modern drug design*. In Hillisch, A. and Hilgenfeld, R. (eds.) *Modern Methods of Drug Discovery*, Birkhäuser, Basel: **2003**, 223–241.

56 RCSB Protein Data Bank, from the Research Collaboratory for Structural Bioinformatics, http://www.rcsb.org/pdb/index.html.

57 Berman, H.M., Westbrook, J., Feng, Z., et al. The protein data bank. Nucleic Acids Res. **2000**, 28: 235–42.

58 Bohacek, R.S., McMartin, C. and Guida, W.C. The art and practice of structure-based drug design: a molecular modeling perspective. Med. Res. Rev. **1996**, 16: 3–50.

59 Kubinyi, H. Structure-based design of enzyme inhibitors and receptor ligands. Curr. Opin. Drug Discov. Dev. **1998**, 1: 4–15.

60 Murcko, M., Caron, P.R. and Charifson, P.S. Structure-based drug design. Annu. Rep. Med. Chem. **1999**, 34: 297–306.

61 Klebe, G. Recent developments in structure-based drug design. J. Mol. Med. **2000**, 78: 269–81.

62 Böhm, H.-J. and Stahl, M. Structure-based library design: molecular modelling merges with combinatorial chemistry. Curr. Opin. Chem. Biol. **2000**, 4: 283–6.

63 Mattos, C. and Ringe, D. Multiple binding modes. In Kubinyi, H. (ed.) *3D-QSAR in Drug Design. Theory, Methods and Applications*, ESCOM, Leiden: **1993**, 226–54.

64 Maignan, S., Guilloteau, J.-P., Choi-Sledeski, et al. Molecular structures of human factor Xa complexed with ketopiperazine inhibitors: preference for a neutral group in the S1 pocket. J. Med. Chem. **2003**, 46: 685–90.

65 Teague, S.J. Implications of protein flexibility for drug discovery. Nat. Rev. Drug Discov. **2003**, 2: 527–41.

66 Reyda, S., Sohn, C., Klebe, G., Rall, K., Ullmann, D., Jakubke, H.D. and Stubbs, M.T. Reconstructing the binding site of factor Xa in trypsin reveals ligand-induced structural plasticity. J. Mol. Biol. **2003**, 325: 963–77.

67 Ladbury, J.E. Just add water! The effect of water on the specificity of protein–ligand binding sites and its potential application to drug design. Chem. Biol. **1996**, 3: 973–80.

68 Stubbs, M.T., Reyda, S., Dullweber, F., Moller, M., Klebe, G., Dorsch, D., Mederski, W.W.K.R. and Wurziger, H. pH-dependent binding modes observed in trypsin crystals: lessons for structure-based drug design. ChemBioChem **2002**, 3: 246–9.

69 Davis, A.M., Teague, S.J. and Kleywegt, G.J. Application and limitations of X-ray crystallographic data in structure-based ligand and drug design. Angew. Chem. Int. Ed. Engl. **2003**, 42: 2718–36.

70 Hooft, R.W.W., Vriend, G., Sander, C. and Abola, E.E. Errors in protein structures. Nature **1996**, 381: 272.

71 Nissink, J.W.M., Murray, C., Hartshorn, M., Verdonk, M.L., Cole, J.C. and Taylor, R. A new test set for validating predictions of protein-ligand interaction. Proteins **2002**, 49: 457–71.

72 Babine, R.E. and Bender, S.L. Molecular recognition of protein-ligand complexes: applications to drug design. Chem. Rev. **1997**, 97: 1359–1472.

73 Gohlke, H. and Klebe, G. Approaches to the description and prediction of the binding affinity of small-molecule ligands to macromolecular receptors. Angew. Chem. Int. Ed. Engl. **2002**, 41: 2644–76.

74 Tame, J.R.H. Scoring functions: a view from the bench. J. Comput. Aided Mol. Des. **1999**, 13: 99–108.

75 Oprea, T.I. and Marshall, G.R. Receptor-based prediction of binding affinities. In Kubinyi, H., Folkers, G.

and Martin, Y.C. (eds.) *3D-QSAR in Drug Design*, Vol. 2, Kluwer, Dordrecht: **1998**, 35–61.

76 Böhm, H.J. and Stahl, M. The use of scoring functions in drug discovery applications. Rev. Comput. Chem. **18**: 41–87.

77 Gohlke, H. and Klebe, G. Statistical potentials and scoring functions applied to protein–ligand binding. Curr. Opin. Struct. Biol. **2001**, 11: 231–5.

78 Charifson, P.S., Corkerey, J.J., Murcko, M.A. and Walters, W.P. Consensus scoring: a method for obtaining improved hit rates from docking databases of three-dimensional structures into proteins. J. Med. Chem. **1999**, 42: 5100–9.

79 Clark, R.D., Strizhev, A., Leonard, J.M., Blake, J.F. and Matthew, J.B. Consensus scoring for ligand/protein interactions. J. Mol. Graph. Mod. **2002**, 20: 281–95.

80 Gohlke, H. and Klebe, G. DrugScore meets comfa: adaptation of fields for molecular comparison (AFMoC) or how to tailor knowledge-based pair-potentials to a particular protein. J. Med. Chem. **2002**, 45: 4153–70.

81 Gohlke, H., Hendlich, M. and Klebe, G. Knowledge-based scoring function to predict protein–ligand interactions. J. Mol. Biol. **2000**, 295: 337–56.

82 Cramer, R.D., Patterson, D.E. and Bunce, J.E. Comparative molecular field analysis (CoMFA). 1. Effect of shape on binding of steroids to carrier proteins. J. Am. Chem. Soc. **1988**, 110: 5959–67.

83 Clark, M., Cramer, R.D., Jones, D.M., Patterson, D.E. and Simeroth, P.E. Comparative molecular field analysis (CoMFA). 2. Towards its use with 3D-structural databases. Tetrahedron. Comput. Methods **1990**, 3: 47–59.

84 (a) Wold, S., Albano, C., Dunn, W.J., et al. Multivariate data analysis in chemistry. In Kowalski, B. (ed.) *Chemometrics: Mathematics and Statistics in Chemistry*, Reidel, Dordrecht: **1984**, 17–95 (b) Dunn, W.J., Wold, S., Edlund, U., Hellberg, S. and Gasteiger, J. Multivariate structure–activity relationships between data from a battery of biological tests and an ensemble of structural descriptors: the PLS method. Quant. Struct.-Act. Relat. **1984**, 3: 131–7.

85 Ortiz, A.R., Pisabarro, M.T., Gago, F. and Wade, R.C. Prediction of drug binding affinities by comparative binding energy analysis. J. Med. Chem. **1995**, 38: 2681–91.

86 Ortiz, A.R., Pastor, M., Palomer, A., Cruciani, G., Gago, F. and Wade, R.C. Reliability of comparative molecular field analysis models: effects of data scaling and variable selection using a set of human synovial fluid phospholipase A2 inhibitors. J. Med. Chem. **1997**, 40: 1136–48.

87 Perez, C., Pastor, M., Ortiz, A.R. and Gago, F. Comparative binding energy analysis of HIV-1 protease inhibitors: incorporation of solvent effects and validation as a powerful tool in receptor-based drug design. J. Med. Chem. **1998**, 41: 836–52.

88 Wang, T. and Wade, R.C. Comparative binding energy (COMBINE) analysis of influenza neuraminidase-inhibitor complexes. J. Med. Chem. **2001**, 44: 961–71.

89 Rognan, D., Lauemoller, S.L., Holm, A., Buus, S. and Tschinke, V. Predicting binding affinities of protein ligands from three-dimensional models: application to peptide binding to Class I major histocompatibility proteins. J. Med. Chem. **1999**, 42: 4650–8.

90 Murray, C.W., Auton, T.R. and Eldridge, M.D. Empirical scoring functions. II. The testing of an empirical scoring function for the prediction of ligand-receptor affinities and the use of Bayesian regression to improve the quality of the model. J. Comput. Aided Mol. Des. **1998**, 12: 503–19.

91 Eldridge, M.D., Murray, C.W., Auton, T.R., Paolini, G.V. and Mee, R.P. Empirical scoring functions: I. The development of a fast empirical scoring function to estimate the binding affinity of ligands in receptor complexes. J. Comput. Aided Mol. Des. **1997**, 11: 425–45.

92 Klebe, G., Abraham, U. and Mietzner, T. Molecular similarity indices in a comparative analysis (CoMSIA) of drug molecules to correlate and predict their biological activity. J. Med. Chem. **1994**, 37: 4130–46.

93 (a) Wold, S. Cross-validatory estimation of the number of component in factor and principal component models. Technometrics **1978**, 4: 397–405; (b) Cramer, R.D., Bunce, J.D. and Patterson, D.E. Crossvalidation, bootstrapping and partial least squares compared with multiple regression in conventional QSAR studies. Quant. Struct.-Act. Relat. **1988**, 18–25.

94 Waller, C.L., Oprea, T.I., Giolitti, A. and Marshall, G.R. Three-dimensional QSAR of human immunodeficiency virus (I) protease inhibitors. 1. A CoMFA study employing experimentally-determined alignment rules. J. Med. Chem. **1993**, 36: 4152–60.

95 Oprea, T.I., Waller, C.L. and Marshall, G.R. Three-dimensional quantitative structure–activity relationship of human immuno-deficiency virus (I) protease inhibitors. 2. Predictive power using limited exploration of alternate binding modes. J. Med. Chem. **1994**, 37: 2206–15.

96 Wei, D.T., Meadows, J.C. and Kellogg, G.E. Effects of entropy on QSAR equations for HIV-1 protease: 1. Using hydropathic binding descriptors. 2. Unrestrained complex structure optimizations. Med. Chem. Res. **1997**, 7: 259–270.

97 Sippl, W. Receptor-based 3D QSAR analysis of estrogen receptor ligands – merging the accuracy of receptor-based alignments with the computational efficiency of ligand-based methods. J. Comput. Aided Mol. Des. **2000**, 14: 559–72.

98 Sippl, W., Contreras, J.M., Parrot, I., Rival, Y.M. and Wermuth, C.G. Structure-based 3D QSAR and design of novel acetylcholinesterase inhibitors. J. Comput. Aided Mol. Des. **2001**, 15: 395–410.

99 Matter, H., Schwab, W., Barbier, D., , et al. Quantitative structure–activity relationship of human neutrophil collagenase (MMP-8) inhibitors using comparative molecular field and X-Ray structure analysis. J. Med. Chem. **1999**, 42: 1908–20.

100 Matter, H. and Schwab, W. Affinity and selectivity of matrix metalloproteinase inhibitors: a chemometrical study from the perspective of ligands and proteins. J. Med. Chem. **1999**, 42: 4506–23.

101 Naumann, T. and Matter, H. Structural classification of protein kinases using 3D molecular interaction field analysis of their ligand binding sites: target family landscapes. J. Med. Chem. **2002**, 45, 2366–78.

102 Matter, H., Defossa, E., Heinelt, U., et al. Design and quantitative structure–activity relationship of 3-amidinobenzyl-1H-indole-2-carboxamides as potent, non-chiral and selective inhibitors of blood coagulation factor Xa. J. Med. Chem. **2002**, 45: 2749–69.

103 Matter, H. Computational approaches towards the quantification of molecular diversity and design of compound libraries. In Hillisch, A. and Hilgenfeld, R. (eds.) *Modern Methods of Drug Discovery*, Birkhäuser, Basel: **2003**, 125–156.

104 Schuffenhauer, A., Zimmermann, J., Stoop, R., Van der Vyver, J.J., Lecchini, S. and Jacoby, E. An ontology for pharmaceutical ligands and its application for *in silico* screening and library design. J. Chem. Inf. Comput. Sci. **2002**, 42: 947–55.

105 Schuffenhauer, A., Floersheim, P., Acklin, P. and Jacoby, E. Similarity metrics for ligands reflecting the similarity of the target proteins. J. Chem. Inf. Comput. Sci. **2003**, 43: 391–405.

106 Wong, G., Koehler, K.F., Skolnick, P., et al. Synthetic and computer-assisted analysis of the structural requirements for selective, high-affinity ligand binding to diazepam-insensitive benzo diazepine receptors. J. Med. Chem. **1993**, 36: 1820–30.

107 Matter, H., Defossa, E., Heinelt, U., Naumann, T., Schreuder, H. and

WILDGOOSE, P. Combining structure-based design and 3D-QSAR towards the discovery of non-chiral, potent and selective factor Xa inhibitors. In *Proc. 13th Eur. Symp. on Quantitative Structure–Activity Relationships*, Barcelona: **2001**, 177–85.

108 KASTENHOLZ, M.A., PASTOR, M., CRUCIANI, G., HAAKSMA, E.E.J. AND FOX, T. GRID/CPCA: a new computational tool to design selective ligands. J. Med. Chem. **2000**, 43: 3033–44.

109 BOEHM, M., STUERZEBECHER, J. AND KLEBE, G. Three-dimensional quantitative structure–activity relationship analyses using comparative molecular field analysis and comparative molecular similarity indices analysis to elucidate selectivity differences of inhibitors binding to trypsin, thrombin, and factor Xa. J. Med. Chem. **1999**, 42: 458–77.

110 CHEN, Q., WU, C., MAXWELL, D., KRUDY, G.A., DIXON, R.A.F. AND YOU, T.J. A 3D QSAR analysis of *in vitro* binding affinity and selectivity of 3-isoxazolylsulfonylaminothiophenes as endothelin receptor antagonists. Quant. Struct.-Act. Relat. **1999**, 18: 124–33.

111 BOSTROM, J., BOEHM, M., GUNDERTOFTE, K. AND KLEBE, G. A 3D QSAR study on a set of dopamine D4 receptor antagonists. J. Chem. Inf. Comput. Sci. **2003**, 43: 1020–7.

112 BALLE, T., ANDERSEN, K., SOBY, K.K. AND LILJEFORS, T. α1 Adrenoceptor subtype selectivity 3D-QSAR models for a new class of α1 adrenoceptor antagonists derived from the novel antipsychotic sertindole. J. Mol. Graph. Mod. **2003**, 21: 523–34.

113 GOODFORD, P. A computational procedure for determining energetically favorable binding sites on biologically important macromolecules. J. Med. Chem. **1985**, 28: 849–57.

114 NISSINK, J.W.M., VERDONK, M.L. AND KLEBE, G. Simple knowledge-based descriptors to predict protein–ligand interactions. Methodology and validation. J. Comput. Aided Mol. Des. **2000**, 14: 787–803.

115 VERDONK, M.L., COLE, J.C., WATSON, P., GILLET, V. AND WILLETT, P. SuperStar: improved knowledge-based interaction fields for protein binding sites. J. Mol. Biol. **2001**, 307: 841–59.

116 GOHLKE, H., HENDLICH, M. AND KLEBE, G. Predicting binding modes, binding affinities and "hot spots" for protein–ligand complexes using a knowledge-based scoring function. Perspect. Drug Discov. Dev. **2000**, 20: 115–44.

117 WESTERHUIS, J.A., KOURTI, T. AND MACGREGOR, J.F. Analysis of multi-block and hierarchical PCA and PLS models. J. Chemometrics **1998**, 12: 301–21.

118 RIDDERSTRÖM, M., ZAMORA, I., FJELLSTRÖ, O. AND ANDERSSON, T.B. Analysis of selective regions in the active sites of human cytochromes P450, 2C8, 2C9, 2C18, and 2C19 homology models using GRID/CPCA. J. Med. Chem. **2001**, 44: 4072–81.

119 WILLIAMS, P.A., COSME, J., SRIDHAR, V., JOHNSON, E.F. AND MCREE, D.E. Mammalian microsomal cytochrome P450 monooxygenase: structural adaptations for membrane binding and functional diversity. Mol. Cell **2000**, 5: 121–31.

120 TERP, G.E., CRUCIANI, G., CHRISTENSEN, I.T. AND JORGENSEN, F.S. Structural differences of matrix metalloproteinases with potential implications for inhibitor selectivity examined by the GRID/CPCA approach. J. Med. Chem. **2002**, 45: 2675–84.

121 MYSHKIN, E. AND WANG, B. Chemometrical classification of ephrin ligands and eph kinases using GRID/CPCA approach. J. Chem. Inf. Comput. Sci. **2003**, 43: 1004–10.

122 KURZ, M., BRACHVOGEL, V., MATTER, H., STENGELIN, S., THÜRING, H. AND KRAMER, W. Insights into the bile acid transportation system the human ileal lipid-binding protein (ILBP) – cholyltaurine complex and its comparison to homologous structures. Proteins **2003**, 50: 312–28.

123 SHERIDAN, R.P., HOLLOWAY, M.K., MCGAUGHEY, G., MOSLEY, R.T. AND SINGH, S.B. A simple method for visualizing the differences between related receptor sites. J. Mol. Graph. Mod. **2002**, 21: 71–9.

124 Matter, H., Naumann, T. and Pirard, B. Target family landscapes to match ligand selectivity with binding site topology in chemical biology. In Ford, M., Livingstone, D., Dearden, J. and van de Waterbeemd, H. (eds.) *EuroQSAR2002: Designing Drugs and Crop Protectants: Processes, Problems and Solutions*, Blackwell, Oxford: **2003**, 183–5.

125 Lapinsh, M., Prusis, P., Gutcaits, A., Lundstedt, T. and Wikberg, J.E.S. Development of proteo-chemometrics: a novel technology for the analysis of drug–receptor interactions. Biochim. Biophys. Acta **2001**, 1525: 180–90.

126 Lapinsh, M., Prusis, P., Mutule, I., Mutulis, F. and Wikberg, J.E.S. QSAR and proteo-chemometric analysis of the interaction of a series of organic compounds with melanocortin receptor subtypes. J. Med. Chem. **2003**, 46: 2572–9.

127 Lapinsh, M., Prusis, P., Lundstedt, T. and Wikberg, J.E.S. Proteochemometrics modeling of the interaction of amine G-protein coupled receptors with a diverse set of ligands. Mol. Pharmacol. **2002**, 61: 1465–75.

128 Ekins, S., Waller, C.L., Swann, P.W., Cruciani, G., Wrighton, S.A. and Wikel, J.H. Progress in predicting human ADME parameters in *silico*. J. Pharm. Toxicol. Methods **2000**, 44: 251–72.

129 Lipinski, C.A., Lombardo, F., Dominy, B.W. and Feeney, P.J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Adv. Drug Deliv. Rev. **2001**, 46: 3–26.

130 Jorgensen, W.L. and Duffy, E.M. Prediction of drug solubility from structure. Adv. Drug Deliv. Rev **2002**, 54: 355–66.

131 Rosenberg, M.F., Kamis, A.B., Callaghan, R., Higgins, C.F. and Ford, R.C. Three-dimensional structures of the mammalian multidrug resistance P-glycoprotein demonstrate major conformational changes in the transmembrane domains upon nucleotide binding. J. Biol. Chem. **2003**, 278: 8294–9.

132 Litman, T., Druley, T.E., Stein, W.D. and Bates, S.E. From MDR to MXR: new understanding of multidrug resistance systems, their properties and clinical significance. Cell. Mol. Life Sci. **2001**, 58: 931–59.

133 Zhao, Y.H., Le, J., Abraham, M.H., et al., Evaluation of human intestinal absorption data and subsequent derivation of a quantita structure–activity relationship (QSAR) with the Abraham descriptors. J. Pharm. Sci. **2001**, 90: 749–84.

134 Fu, X.C., Liang, W.Q. and Yu, Q.S. Correlation of drug absorption with molecular charge distribution. Pharmazie **2001**, 56: 267–8.

135 Agatonovich-Kustrin, S., Beresford, R. and Yusof, A.P.M. ANN modeling of the penetration across a polydimethylsiloxane membrane from theoretically derived molecular descriptors. J. Pharm. Biomed. Anal. **2001**, 25: 227–37.

136 Norinder, U. and Österberg, T. Theoretical calculation and prediction of drug transport processes using simple parameters and partial least squares projections to latent structures (PLS) statistics. The use of electrotopological state indices. J. Pharm. Sci. **2001**, 90: 1076–85.

137 Agoram, B., Woltosz, W.S. and Bolger, M.B. Predicting the impact of physiological and biochemical processes on oral drug bioavailability. Adv. Drug Deliv. Rev. **2001**, 90: S41–67.

138 Norris, D.A., Leesman, G.D., Sinko, P.J. and Grass, G.M. Development of predictive pharmacokinetic simulation models for drug discovery. J. Control Relat. **2000**, 65: 55–62.

139 Parrott, N. and Lavé, T. Prediction of intestinal absorption: comparative assessment of Gastroplus and Idea. Eur. J. Pharm. Sci. **2002**, 17: 51–61.

140 Kariv, I., Rourick, R.A., Kassel, D.B. and Chung, T.D.Y. Improvement of "hit-to-lead" optimization by integration of *in vitro* HTS experimental models for early determination of pharmacokinetic properties. Comb. Chem. High Throughput Screen. **2002**, 5: 459–72.

141 Beresford, A.P., Selick, H.E. and Tarbit, M.H. The emerging importance

of predictive ADME simulation in drug discovery. Drug Discov. Today **2002**, 7: 109–16.

142 YOSHIDA, F. AND TOPLISS, J.G. QSAR model for drug human oral bioavailability. J. Med. Chem. **2000**, 43: 2575–85.

143 MADAN, A., USUKI, E., BURTON, L.A., OGILVIE, B.W. AND PARKINSON, A. *In vitro* approaches for studying the inhibition of drug-metabolizing enzymes and identifying the drug-metabolizing enzymes responsible for the metabolism of drugs. In RODRIGUEZ, A.D. (ed.) *Drug Drug Interaction*. Marcel Dekker, New York, NY: **2002**, 217–94.

144 DE GROOT, M.J., ALEX, A.A. AND JONES, B.C. Development of a combined protein and pharmacophore model for cytochrome P450 2C9. J. Med. Chem. **2002**, 45: 1983–93.

145 BIDWELL, L.M., MCMANUS, M., GAEDIGK, A., KAKUTA, Y., NEGISHI, M., PEDERSEN, L. AND MARTIN, J.L. Crystal structure of human catecholamine sulfotransferase. J. Mol. Biol. **1999**, 293: 521–30.

146 EKINS, S., DE GROOT, M. AND JONES, J.P. Pharmacophore and three-dimensional quantitative structure activity relationship methods for modeling cytochrome P450 active sites. Drug Metab. Dispos. **2001**, 29: 936–44.

147 HIGGINS, L., KORZEKWA, K.R., RAO, S., SHOU, M. AND JONES, J.P. An assessment of the reaction energetics for cytochrome P450-mediated reactions. Arch. Biochem. Biophys. **2001**, 385: 220–30.

148 SINGH, S.B., SHEN, L.Q., WALKER, M.J. AND SHERIDAN, R.P. A model for predicting likely sites of CYP3A4-mediated metabolism on drug-like molecules. J. Med. Chem. **2003**, 46: 1330–6.

149 SHEN, M., XIAO, Y., GOLBRAIKH, A., GOMBAR, V.K. AND TROPSHA, A. Development and validation of k-nearest-neighbor QSPR models of metabolic stability of drug candidates. J. Med. Chem. **2003**, 46: 3013–20.

150 LANGOWSKI, J. AND LONG, A. Computer systems for the prediction of xenobiotic metabolism. Adv. Drug Deliv. Rev. **2002**, 54: 407–15.

151 GREENE, N. Computer systems for the prediction of toxicity: an update. Adv. Drug Deliv. Rev. **2002**, 54: 417–31.

152 VANDENBERG, J.I., WALKER, B.D. AND CAMPBELL, T.J. HERG K^+ channels: friend and foe. Trends Pharmacol. Sci. **2001**, 22: 240–6.

153 VAN DE WATERBEEMD, H., SMITH, D.A., BEAUMONT, K. AND WALKER, D.K. Property-based design: optimization of drug absorption and pharmacokinetics. J. Med. Chem. **2001**, 44: 1313–33.

154 VAN DE WATERBEEMD, H. AND GIFFORD, E. ADMET *in silico* modelling: towards prediction paradise? Nat. Drug Discov. **2003**, 2: 192–204.

155 CRUCIANI, G., CRIVORI, P., CARRUPT, P.A. AND TESTA, B. Molecular fields in quantitative structure–permeation relationships: the VolSurf approach. THEOCHEM **2000**, 503: 17–30.

156 CRUCIANI, G., PASTOR, M. AND GUBA, W. VolSurf: a new tool for the pharmacokinetic optimization of lead compounds. Eur. J. Pharm. Sci. **2000**, 11: S29–39.

157 GUBA, W. AND CRUCIANI, G. Molecular field-derived descriptors for the multivariate modelling of pharmacokinetic data. In GUNDERTOFTE, K. AND JORGENSEN, F.S. (eds.) *Molecular Modelling and Prediction of Bioactivity, Proceedings of the 12th European Symposium on Quantitative Structure–Activity Relationships (QSAR'98)*, Plenum Press, New York, NY: **2000**, 89–95.

158 ALIFRANGIS, L.H., CHRISTENSEN, I.T., BERGLUND, A., SANDBERG, M., HOVGAARD, L. AND FROKJAER, S. Structure–property model for membrane partitioning of oligopeptides. J. Med. Chem. **2000**, 43: 103–13.

159 CRIVORI, P., CRUCIANI, G., CARRUPT, P.A. AND TESTA, B. Predicting blood-brain barrier permeation from three-dimensional molecular structure. J. Med. Chem. **2000**, 43: 2204–16.

160 PFEIFFER-MAREK, S., MATTER, H., ENGLERT, H., GERLACH, U., KNIEPS, S., SCHUDOK, M. AND LEHR, K.H. Using VolSurf descriptors to model pharmacokinetic properties during lead optimization. In FORD, M., LIVINGSTONE, D., DEARDEN, J. AND VAN DE WATERBEEMD, H. (eds.) *EuroQSAR 2002: Designing Drugs and Crop Protectants: Processes, Problems and Solutions.* Blackwell, Oxford: **2003**, 104–5.

161 COLMENAREJO, G., ALVAREZ-PEDRAGLIO, A. AND LAVANDERA, J.-L. Cheminformatic models to predict binding affinities to human serum albumin. J. Med. Chem. **2001**, 44: 4370–8.

162 PETITPAS, I., BHATTACHARYA, A.A., TWINE, S., EAST, M. AND CURRY, S. Crystal structure analysis of warfarin binding to human serum albumin. Anatomy of drug site I. J. Biol. Chem. **2001**, 276: 22804–9.

163 ZHU, C., JIANG, L., CHEN, T.M. AND HWANG, K.K. A comparative study of artificial membrane permeability assay for high throughput profiling of drug absorption potential. Eur. J. Med. Chem. **2002**, 37: 399–407.

164 SALMINEN, T., PULLI, A. AND TASKINEN, J. Relationship between immobilised artificial membrane chromatographic retention and the brain penetration of structurally diverse drugs. J. Pharm. Biomed. Anal. **1997**, 15: 469–77.

165 IRVINE, J.D., LOCKHART, L.T., CHEONG, J., TOLAN, J.W., SELICK, H.E. AND GROVE, J.R. MDCK (Madin-Darby canine kidney) cells: A tool for membrane permeability screening. J. Pharm. Sci. **1999**, 88: 28–33.

166 YAZDANIAN, M., GLYNN, S.L., WRIGHT, J.L. AND HAWI, A. Correlating partitioning and Caco-2 cell permeability of structurally diverse small molecular weight compounds. Pharm. Res. **1998**, 15: 1490–4.

167 DOYLE, D.A., MORAIS CABRAL, J., PFUETZNER, R.A., KUO, A., GULBIS, J.M., COHEN, S.L., CHAIT, B.T. AND MACKINNON, R. The structure of the potassium channel: molecular basis of K^+ conduction and selectivity. Science **1998**, 280: 69–77.

168 MITCHESON, J.S., CHEN, J., LIN, M., CULBERSON, C. AND SANGUINETTI, M.C. A structural basis for drug-induced long QT syndrome. Proc. Natl Acad. Sci. USA **2000**, 97: 12329–333.

169 PEARLSTEIN, R., VAZ, R. AND RAMPE, D. Understanding the structure–activity relationship of the human ether-a-go-go-related gene cardiac K^+ channel. A model for bad behavior. J. Med. Chem. **2003**, 46: 2017–22.

170 ROCHE, O., TRUBE, G., ZUEGGE, J., PFLIMLIN, P., ALANINE, A. AND SCHNEIDER, G. A virtual screening method for prediction of the hERG potassium channel liability of compound libraries. ChemBioChem **2002**, 3: 455–9.

171 CAVALLI, A., POLUZZI, E., DE PONTI, F. AND RECANATINI, M. Toward a pharmacophore for drugs inducing the long QT syndrome: insights from a CoMFA study of HERG K^+ channel blockers. J. Med. Chem. **2002**, 45: 3844–53.

172 MATTER, H. Unpublished results.

173 PEARLSTEIN, R.A., VAZ, R.J., KANG, J., ET AL., Characterization of HERG potassium channel inhibition using CoMSiA 3D QSAR and homology modeling approaches. Bioorg. Med. Chem. Lett. **2003**, 13: 1829–35.

174 JIANG, Y., LEE, A., CHEN, J., CADENE, M., CHAIT, B.T. AND MACKINNON, R. Crystal structure and mechanism of a calcium-gated potassium channel. Nature **2002**, 417: 515–22.

175 FRIESEN, R.W., DUCHARME, Y., BALL, R.G., ET AL., Optimization of a tertiary alcohol series of phosphodiesterase-4 (PDE4) inhibitors: structure–activity relationship related to PDE4 inhibition and human ether-a-go-go related gene potassium channel binding affinity. J. Med. Chem. **2003**, 46: 2413–26.

176 STRESSER, D.M., BLANCHARD, A.P., TURNER, S.D., ERVE, J.C.L., DANDENEAU, A.A., MILLER, V.P. AND CRESPI, C.L. Substrate-dependent modulation of CYP3A4 catalytic activity: analysis of 27 test compounds with four fluorometric substrates. Drug Metab. Dispos. **2000**, 28: 1440–8.

177 LEWIS, D.F.V. Structural characteristics of human P450s involved in drug metabolism: QSARs and lipophilicity profiles. Toxicology **2000**, 144: 197–203.

178 Lewis, D.F.V. Essential requirements for substrate binding affinity and selectivity toward human CYP2 family enzymes. Arch. Biochem. Biophys. **2003**, 409: 32–44.

179 Ekins, S., Bravi, G., Ring, B.J., Gillespie, T.A., Gillespie, J.S., Vamdenbranden, M., Wrighton, S.A. and Wikel, J.H. Three-dimensional quantitative structure activity relationship analyses of substrates for CYP2B6. J. Pharmacol. Exp. Ther. **1999**, 288: 21–9.

180 Ekins, S., Bravi, G., Wikel, J. and Wrighton, S.A. Three-dimensional quantitative structure activity relationship analysis of cytochrome P-450 3A4 substrates. J. Pharmacol. Exp. Ther. **1999**, 291: 424–33.

181 Ekins, S., Bravi, G., Binkley, S., Gillespie, J.S., Ring, B.J., Wikel, J.H. and Wrighton, S.A. Three- and four-dimensional quantitative structure activity relationship analyses of cytochrome P-450 3A4 inhibitors. J. Pharmacol. Exp. Ther. **1999**, 290: 429–38.

182 Ekins, S., Bravi, G., Binkley, S., Gillespie, J.S., Ring, B.J., Wikel, J.H. and Wrighton, S.A. Three- and four-dimensional-quantitative structure activity relationship (3D/4D-QSAR) analyses of CYP2C9 inhibitors. Drug Metab. Dispos. **2000**, 28: 994–1002.

183 Ekins, S., Bravi, G., Binkley, S., Gillespie, J.S., Ring, B.J., Wikel, J.H. and Wrighton, S.A. Three and four dimensional-quantitative structure activity relationship (3D/4D-QSAR) analyses of CYP2D6 inhibitors. Pharmacogenetics **1999**, 9: 477–89.

184 Rao, S., Aoyama, R., Schrag, M., Trager, W.F., Rettie, A. and Jones, J.F. A refined 3-dimensional QSAR of cytochrome P450 2C9: computational predictions of drug interactions. J. Med. Chem. **2000**, 43: 2789–96.

185 Afzelius, L., Masimirembwa, C.M., Karlen, A., Andersson, T.B. and Zamora, I. Discriminant and quantitative PLS analysis of competitive CYP2C9 inhibitors versus non-inhibitors using alignment independent GRIND descriptors. J. Comput. Aided Mol. Des. **2002**, 16: 443–58.

186 Pastor, M., Cruciani, G., McLay, I., Pickett, S. and Clementi, S. GRid-INdependent descriptors (GRIND): a novel class of alignment-independent three-dimensional molecular descriptors. J. Med. Chem. **2000**, 43: 3233–43.

187 De Groot, M.J., Ackland, M.J., Horne, V.A., Alex, A.A. and Jones, B.C. Novel approach to predicting P450-mediated drug metabolism: development of a combined protein and pharmacophore model for CYP2D6. J. Med. Chem. **1999**, 42: 1515–24.

188 De Groot, M.J., Ackland, M.J., Horne, V.A., Alex, A.A. and Jones, B.C. A novel approach to predicting P450 mediated drug metabolism. CYP2D6 catalyzed N-dealkylation reactions and qualitative metabolite predictions using a combined protein and pharmacophore model for CYP2D6. J. Med. Chem. **1999**, 42: 4062–70.

189 Zamora, I., Afzelius, L. and Cruciani, G. Predicting drug metabolism: a site of metabolism prediction tool applied to the cytochrome P450 2C9. J. Med. Chem. **2003**, 46: 2313–24.

190 Wester, M.R., Johnson, E.F., Marques-Soares, C., Dijols, S., Dansette, P.M., Mansuy, D. and Stout, C.D. Structure of mammalian cytochrome P450 2C5 complexed with diclofenac at 2.1-Å resolution: evidence for an induced fit model of substrate binding. Biochemistry **2003**, 42: 9335–45

191 Wester, M.R., Johnson, E.F., Marques-Soares, C., Dansette, P.M., Mansuy, D. and Stout, C.D. Structure of a substrate complex of mammalian cytochrome P450 2C5 at 2.3-Å resolution: evidence for multiple substrate binding modes. Biochemistry **2003**, 42: 6370–9.

192 Williams, P.A., Cosme, J., Ward, A., Angove, H.C., Vinkovic, D.M. and Jhoti, H. Crystal structure of human cytochrome P450 2C9 with bound warfarin. Nature **2003**, 424: 464–8.

193 Hellberg, S., Eriksson, L., Jonsson, J., Lindgren, F., Sjöström, M., Skagerberg, B., Wold, S. and Andrews, P. Minimum analog peptide

sets (MAPS) for quantitative structure–activity relationships. Int. J. Peptide Protein Res. **1991**, 37: 414–24.

194 Atterwill, C.K. and Wing, M.G. *In vitro* preclinical lead optimization technologies (PLOTs) in pharmaceutical development. Toxicol. Lett. **2002**, 127: 143–51.

195 Linusson, A., Gottfries, J., Olsson, T., Oernskov, E., Folestad, S., Norden, B. and Wold, S. Statistical molecular design, parallel synthesis, and biological evaluation of a library of thrombin inhibitors. J. Med. Chem. **2001**, 44: 3424–39.

196 Eriksson, U.G., Bredberg, U., Gislen, K., Johansson, L.C., Frison, L., Ahnoff, M. and Gustafsson, D. Pharmacokinetics and pharmacodynamics of ximelagatran, a novel oral direct thrombin inhibitor, in young healthy male subjects. Eur. J. Clin. Pharmacol. **2003**, 59: 35–43.

197 Sugano, K., Yoshida, S., Takaku, M., Haramura, M., Saitoh, R., Nabuchi, Y. and Ushio, H. Quantitative structure–intestinal permeability relationship of benzamidine analogue thrombin inhibitor. Bioorg. Med. Chem. Lett. **2000**, 10: 1939–42.

198 Burgey, C.S., Robinson, K.A., Lyle, T.A., et al. Metabolism-directed optimization of 3-aminopyrazinone acetamide thrombin inhibitors. Development of an orally bioavailable series containing P1 and P3 pyridines. J. Med. Chem. **2003**, 46: 461–73.

199 Hasegawa, K., Shindoh, H., Shiratori, Y., Ohtsuka, T., AOKI, Y., Ichihara, S., Horii, I. and Shimma, N. Cassette dosing approach and quantitative structure–pharmacokinetic relationship study of antifungal N-myristoyl- transferase inhibitors. J. Chem. Inf. Comput. Sci. **2002**, 42: 968–75.

200 McKenna, J.M., Halley, F., Souness, J.E., McLay, I.M., Pickett, S.D., Collis, A.J., Page, K. and Ahmed, I. An algorithm-directed two component library synthesized via solid-phase methodology yielding potent and orally bioavailable p38 MAP kinase inhibitors. J. Med. Chem. **2002**, 45: 2173–84.

201 Pickett, S.D., McLay, I.M. and Clark, D.E. Enhancing the hit-to-lead properties of lead optimization libraries. J. Chem. Inf. Comput. Sci. **2000**, 40: 263–72.

202 Choi-Sledeski, Y.M., Kearney, R., Poli, G., et al., Discovery of an orally efficacious inhibitor of coagulation factor Xa which incorporates a neutral P1 ligand. J. Med. Chem. **2003**, 46: 681–4.

203 Tucker, T.J., Brady, S.F., Lumma, W.C., et al. Design and synthesis of a series of potent and orally bioavailable noncovalent thrombin inhibitors that utilize nonbasic groups in the P1 position. J. Med. Chem. **1998**, 41: 310–9.

204 Matter, H., Schudok, M., Schwab, W., et al. Tetrahydroisoquinoline-3-carboxylate based matrix metalloproteinase inhibitors: design, synthesis and structure–activity relationship. Bioorg. Med. Chem. **2002**, 10: 3529–44.

205 Baringhaus, K.H., Matter, H., Stengelin, S. and Kramer, W. Substrate specificity of the ileal and the hepatic Na+/bile acid cotransporters of the rabbit. II. A reliable 3D QSAR pharmacophore model for the ileal Na^+/bile acid cotransporter. J. Lipid Res. **1999**, 40: 2158–68.

Note:
This chapter is based on an article published previously in “Chemoinformatics in Drug Discovery”, edited by Tudor I. Oprea (ISBN 978-3-527-30753-1).

Part 6 Molecular Networks

20
Modeling and Simulating Metabolic Networks

Stefan Schuster and David Fell

1 Introduction

This chapter gives an overview of various modeling and simulation techniques in the theoretical description of metabolic (biochemical) networks. While the term "modeling" refers to the establishment of a formal representation of the system or process under study, "simulation" stands for the numerical calculations performed with the model. As most biochemical systems subsist in stable stationary states, the modeling of such states is central to the field. However, oscillatory and other dynamics can also be modeled. First, the fundamentals are outlined: stoichiometry, balance equations and enzyme kinetics. Thereafter, the principles of metabolic pathway analysis are presented. Basic concepts such as conservation relations, nullspace, elementary modes and extreme pathways are explained. The added value of elementary-modes analysis by avoiding incorrect interpretations from graphic representations and by deducing novel interpretations of metabolism is illustrated by examples from carbon metabolism, e.g. in Crassulacean plants. Methods for constructing and using dynamic models of metabolic networks are outlined. Such models can often be simplified by the quasi-equilibrium and quasi-steady-state approximations. As an example of medium-scale analysis, we discuss the modeling of red blood cell metabolism. An outlook on current trends in the field concludes the chapter.

Bioinformatics – From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

2 Fundamentals

2.1 Motivation

The phenomenon of life is tightly coupled to chemical reactions – autotrophic organisms fix CO_2 and convert it to organic compounds, while heterotrophic organisms degrade some of the organic compounds of their food to obtain energy and convert other constituents of the food into organism-specific compounds. At the level of living cells, this chemical transformation of substances is called metabolism (from the Greek: *meta* = within and *bole* = throwing). Most biochemical reactions are catalyzed by specific proteins – enzymes. Technologically relevant metabolic processes include the synthesis of antibiotics, dyes and perfumes, and the production of ethanol by yeast and of amino acids (in large quantities for food supplements) by bacteria. The understanding of these processes is, hence, of great economic importance. Moreover, it has medical relevance because many inherited single-gene diseases result from complete or partial enzyme deficiencies. For example, the complete lack of the enzyme phenylalanine 4-monooxygenase prevents formation of the amino acid tyrosine from the amino acid phenylalanine. This causes the disease phenylketonuria implying mental retardation, due to excessive accumulation of phenylalanine (e.g. Ref. [138]). The disease can be treated by a life-long, nearly phenylalanine-free diet. Partial lack of one or more enzymes causes the complex group of diseases known as mitochondrial cytopathies, one of which, for example, leads to progressive weakness and degeneration of skeletal muscles (ragged red muscle fiber disease). There are only palliative treatments for these disorders (e.g. Ref. [75]).

The metabolism in living cells is a very complex system due to the large number of reactions most of which convert more than one *reactant* (or *substrate*) into more than one *product*. Therefore, a graphical representation in which each reaction is an edge and each substance is a node leads to a network that is more complex than a graph in graph theory, where each edge would only connect two nodes (see also Chapter 43). Importantly, the metabolism of each cell forms a single, interconnected network. For plants and many microbes, the reason is obvious – they can create all their cell contents from one source of carbon. Although many metabolites have only one producing reaction and one consuming reaction, a significant minority are involved in large numbers of reactions. In fact, there appears to be a power-law distribution of connectivities of metabolites [60, 148], so even though the reason for this distribution remains controversial, metabolism forms a highly interconnected network. In addition to the conversion reactions, there are regulatory interactions superimposed on this network, where metabolites activate or inhibit the rates of reactions other than the ones in which they directly participate. As will be explained below, the velocities of biochemical reactions are usually nonlinear functions of the concentrations of substances (substrates, products and *effectors* – the general term for both activators and inhibitors).

For all these reasons, the dynamic behavior of metabolic systems cannot normally be understood intuitively – a mathematical description is needed, often called a model. In fact, nonintuitive behavior can arise in small networks with linear kinetics, as demonstrated by Braess' paradox [9], originally formulated in the context of traffic flows although it also arises in computer networks. In essence, the paradox demonstrates that the addition of a new link in the network in order to increase its capacity can actually lead to a decrease in the average transit velocity.

A model is a simplified representation of some observed system, process or feature. While the term *modeling* refers to the establishment of a formal representation of the system or process under study, *simulation* stands for the numerical calculations performed with the model. For a more detailed discussion of modeling and theory in biochemistry, see Ref. [51].

Mathematical modeling in biochemistry has traditionally focused on the description of stationary states and time courses by using kinetic models (e.g. Refs. [19, 32, 51]. Such models are aimed at predicting the system's dynamics on the basis of the knowledge of the structure in terms of reactions (see Section 2.2) of the network and the kinetic parameters of enzymes (see Section 2.4). Within the last 15 years, however, the focus has somewhat shifted, at least for large-scale models of metabolism, towards network modeling (e.g. Refs. [51, 57, 79, 92, 100, 129]), sometimes called *constraint-based modeling*. The main idea is to leave aside the many kinetic details, which are often unknown, and consider some basic constraints arising from the network structure and thermodynamic principles.

2.2 Stoichiometry

The stoichiometry of a reaction expresses the proportions of changes in mole numbers of the substances involved. For example, in the reaction $2H_2O_2 \rightarrow 2H_2O + O_2$ (catalyzed by the enzyme catalase) the proportion between consumption of hydrogen peroxide and production of oxygen is 2:1. This is expressed by the stoichiometric coefficients 2 and 1. To make a distinction between the substances on the left-hand side of the equation (in chemistry called reactants, in biochemistry often called substrates) and the substances on the right-hand side (products), the stoichiometric coefficients are taken to be negative for substrates and positive for products. Formally, one can write the reaction equation as $2H_2O + O_2 - 2H_2O_2 = 0$.

Of course, most metabolic models comprise more than one biochemical reaction. A concise way of representing the stoichiometric coefficients is to compile them in a matrix. In this matrix, usually rows correspond to metabolites and columns correspond to reactions. The matrix entries signify the multiplicities with which the respective metabolites occur in the respective reactions. The signs indicate the side of the reaction on which the metabolites occur. Consider, for example, the phosphorylation and subsequent isomerization of glucose:

$$\text{glucose} + \text{ATP} \leftrightarrow \text{glucose-6-phosphate} + \text{ADP} \tag{1a}$$

$$\text{glucose-6-phosphate} \leftrightarrow \text{fructose-6-phosphate} \tag{1b}$$

catalyzed by the enzymes hexokinase and phosphoglucoisomerase, respectively. These two reactions are part of the reaction system shown in Figure 1. The stoichiometry matrix corresponding to the above two reactions reads:

$$\mathbf{N} = \begin{pmatrix} -1 & 0 \\ -1 & 0 \\ 1 & -1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}. \tag{2}$$

with the first and second column corresponding to hexokinase and phosphoglucoisomerase, respectively, and the rows (from top to bottom) correspond-

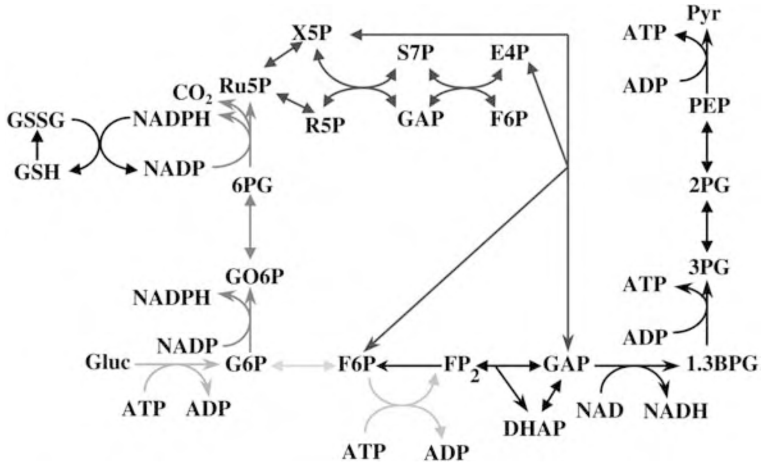


Figure 1 Part of sugar metabolism in various cell types. Abbreviations of metabolite names used in the text: F6P, fructose-6-phosphate; Gluc, glucose; G6P, glucose-6-phosphate, GSH/GSSG, reduced and oxidized forms of glutathione, respectively; Pyr, pyruvate. Red, hexokinase; green, phosphoglucoisomerase; orange, phosphofructokinase; blue (brown), reactions of oxidative (nonoxidative) pentose phosphate pathway.

ing to glucose, ATP, glucose-6-phosphate, ADP and fructose-6-phosphate. Since most substances are involved in only a few reactions, stoichiometric matrices are normally sparse, i.e. they involve a large number of zero entries.

2.3 Balance Equations

Metabolic systems are amenable to mathematical modeling because the law of mass conservation imposes some restrictions that can be expressed in very useful equations. In other processes, this law is also valid, but it does not play such a major role, e.g. in signal transduction systems, the flow of information (which is the focus of interest and which is not subject to conservation laws) is not coupled to the flow of mass (see Chapter 22).

Intuitively speaking, mass conservation implies that the temporal change in the concentration of each metabolite equals the sum of all reaction rates producing that metabolite minus all reaction rates consuming it. Using the stoichiometric coefficients, this can be written as:

$$\frac{\mathrm{d}S_i}{\mathrm{d}t} = \sum_{j=1}^{r} n_{ij} v_j \, , \tag{3}$$

where n_{ij} are the entries in the i-th row and j-th column of matrix $\mathbf{N}$. The symbols S_i and v_j stand for the concentration of the i-th metabolite, S_i, and the rate of the j-th reaction, respectively. r denotes the number of reactions. By gathering the S_i and v_j into vectors $\mathbf{S}$ and $\mathbf{v}$, respectively, Eq. (3) can concisely be written as:

$$\frac{\mathrm{d}\mathbf{S}}{\mathrm{d}t} = \mathbf{N}\mathbf{v} \, . \tag{4}$$

In order to apply this equation in a meaningful way, a distinction should be made between two types of metabolites. *Internal metabolites* are those substances for which the dynamics of their concentrations can be described by Eq. (4) because they are internal to the model. As each model needs to be delimited somehow, some metabolites should form its *boundaries*, i.e. they participate both in reactions involved in the model and in additional (external) reactions. This, however, implies that these substances, which are usually called *external metabolites*, are not properly described by the respective component equations in Eq. (4). Therefore, the rows corresponding to external metabolites should be deleted from the stoichiometry matrix. The distinction between external and internal metabolites is meaningful to properly describe that biochemical networks are open systems, which exhibit a throughput of mass. In the glucose example given above (Eq. 1), all substances except glucose-6-phosphate take part in only one reaction of the model. In the living cell, however, they also participate in other reactions. Therefore, for this small didactic example, it is sensible to consider only glucose-6-phosphate as an internal metabolite. Thus, the stoichiometry matrix reduces to:

$$\mathbf{N} = (1 \ \ -1). \tag{5}$$

This allows one to make assertions about the steady state of the system (see Section 3.2).

2.4 Enzyme Kinetics

To be able to solve the differential equation system (4) analytically or, in most cases, numerically, the right-hand sides should also be expressed in terms of concentrations. This is feasible because the reaction rates do depend on the concentrations, $\mathbf{v} = \mathbf{v}(\mathbf{S})$. It is intuitively clear that the number of molecules converted in a reaction per time depends on the number of molecules present. This dependence is called the *rate law* or *reaction kinetics*. If the reaction is catalyzed by an enzyme, the term *enzyme kinetics* can be used (this term is also used for the entire discipline dealing with such rate laws.) The best known enzyme kinetics is a rate law called after the biochemists Michaelis and Menten [82]:

$$v = \frac{V_{\mathrm{max}} S}{K_{\mathrm{m}} + S}\,, \tag{6}$$

where V_{max} is the limiting rate (formerly maximal activity) and K_{m} is the Michaelis–Menten constant. It describes a unidirectional, monomolecular reaction, i.e. the conversion of one substance, S, into one other substance, P, without a reverse reaction. The italic symbol S denotes the concentration of S. This rate law can be understood very easily: for low substrate concentrations, it simplifies to a linear rate law, $v = V_{\mathrm{max}}/K_{\mathrm{m}} \times S$, which coincides with the usual mass action kinetics known from chemistry. For larger substrate concentrations, the reaction rate increases, but less than linearly, and for very high substrate levels, the reaction rate tends to an asymptotic value, the limiting rate, V_{max}. This is because the rate of formation of the product is proportional to the concentration of the enzyme-substrate complex (ES), which cannot exceed the total amount of the enzyme. This phenomenon is called saturation.

For bidirectional, monomolecular reactions, the Michaelis–Menten kinetics can be generalized to give (e.g. Refs. [19,124]):

$$v = \frac{V_{\mathrm{max}}^{+} S/K_{\mathrm{mS}} - V_{\mathrm{max}}^{-} P/K_{\mathrm{mP}}}{1 + S/K_{\mathrm{mS}} + P/K_{\mathrm{mP}}} \tag{7}$$

where V_{max}^{+} and V_{max}^{-} are the limiting rates of the substrate and product, respectively, and K_{mS} and K_{mP} are the Michaelis–Menten constants of the forward and backward rates, respectively. Bidirectionality means that a forward

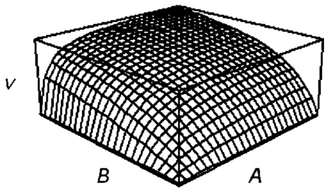


Figure 2 A 3-D plot of the reaction rate of an irreversible, bimolecular enzyme reaction versus the two substrate concentrations, A and B, according to the rate law Eq. (8). Note that any cross-section parallel to the A-axis (or B-axis) yields the plot of a standard Michaelis–Menten rate law (6) with respect to A (respectively B). Parameter values: $V_{\text{max}} = 2$, $K_{\text{iA}} = 1$, $K_{\text{mB}} = 0.5$, $K_{\text{mA}} = 0$.

reaction and a backward reaction proceed simultaneously. In enzyme kinetics, unidirectionality and bidirectionality of reactions are often equated with irreversibility and reversibility, respectively. However, in other applications, such as in network analysis (see Section 3), a reaction is called *irreversible* if it has a fixed net direction while it is allowed to have a backward reaction "underneath". A reaction is called *reversible* if the direction of the net flux can be reversed by changing substrate and/or product concentrations appropriately, although the boundary between irreversible and reversible is arbitrary (see Ref. [32] for a discussion).

Most biochemical reactions are not, however, monomolecular. Also, for these cases, Michaelis–Menten-type rate laws have been derived. Here, we give the rate law for an irreversible reaction with two substrates, A and B (e.g. Refs. [19,124]):

$$v = \frac{V_{\text{max}} \cdot A \cdot B}{K_{\text{iA}} \cdot K_{\text{mB}} + K_{\text{mB}} \cdot A + K_{\text{mA}} \cdot B + A \cdot B} \,. \tag{8}$$

The symbol K with different subscripts stands for phenomenological parameters depending on the rate constants of the elementary steps of enzyme catalysis. If Eq. (8) is derived by the quasi-equilibrium approximation, these parameters are dissociation constants of the enzyme–substrate complexes. For example, K_{mB} is the constant of dissociation of the complex EAB into EA and B. If Eq. (8) is derived by the quasi-steady-state approximation, the dependence is more complicated. In Figure 2, the dependence of the reaction rate on substrate concentrations according to Eq. (8) is plotted.

The various enzyme kinetic rate laws can be derived from the mass-action kinetic equations describing the elementary steps (such as the binding of substrate to the free enzyme). For the monomolecular reaction S → P, the scheme can be written as:

$$\text{E} + \text{S} \rightarrow \text{ES} \rightarrow \text{E} + \text{P}. \tag{9}$$

In the derivation, approximations need to be used, the most common being the quasi-steady-state approximation saying that the concentration of the enzyme–substrate complex is nearly constant in time. A plethora of enzyme kinetic rate laws have been derived, describing, among others, cooperativity, competitive, allosteric and uncompetitive inhibition, various forms of activation, etc. For monographs on enzyme kinetics, the reader is referred to Refs. [19, 124]. Cooperativity means that two or more substrate molecules have to bind to initiate the reaction. This leads to a sigmoidal curve of the rate law (e.g. Hill kinetics and Monod–Wyman–Changeux kinetics), i.e. the curve goes up sharply near an inflection point.

Interestingly, the various versions of Michaelis–Menten kinetics describe the respective enzyme reactions over the entire range of possible substrate concentrations (from zero through infinity) fairly well, provided the equation has been derived for a mechanism appropriate to the enzyme under consideration. In fact, the main motivation for deriving such equations was to diagnose enzyme mechanisms, not so much to describe the enzyme rate function. Such mechanistically derived equations contain many parameters whose values are rarely experimentally known. On the other hand, it is known from sensitivity analysis of model behavior that uncertainty about their exact value is unlikely to have much impact on simulations, so the equations are "over-specified" for use in modeling of cellular situations where the variation in concentration of metabolites is much less than in *in vitro* enzyme kinetics experiments. For this reason, it has been common to use simplified versions of the equations with fewer parameters [14, 156]. There is even a kinetic equation that copes with substrate, product and effector cooperativity, for which there is no satisfactory mechanistically derived justification [53]. In the same vein, simplified rate laws have been proposed [86, 107, 157], which render the differential equation system easier to handle analytically, but do not describe the reaction rate properly at very low or very high substrate concentrations.

3 Network Analysis

3.1 Conservation Relations

From physics, it is known that many systems exhibit so-called conservation quantities, such as energy, momentum and angular momentum. Conservation quantities occur also in many metabolic systems. For example, in Eq. (9) describing the elementary reactions of a monomolecular enzyme reaction, it can be seen that the sum of the concentrations of E and ES is constant in time because this is the total enzyme, which is neither consumed nor produced by the overall catalytic reaction. Of course, in the living cell, the total amount of an enzyme does change, due to gene expression and protein degradation. However, these processes are separate from, and much slower than, the catalytic steps given above and can, therefore, often be neglected. Mathematically, we can describe reaction (9) by the stoichiometry matrix:

$$\mathbf{N} = \begin{pmatrix} -1 & 1 \\ -1 & 0 \\ 1 & -1 \\ 0 & 1 \end{pmatrix} \tag{10}$$

with the rows corresponding to E, S, ES and P (in this order). The sum of the first and third rows (corresponding to E and ES) is the null vector (and so is the sum of the second, third and fourth rows). This implies $\mathrm{d}E/\mathrm{d}t = -\mathrm{d}ES/\mathrm{d}t$, which means that $\mathrm{d}(E + ES)/\mathrm{d}t = 0$, showing that the reaction scheme can never change the total enzyme concentration.

Hence, more generally, conservation relations can be derived from the stoichiometry matrix by analyzing the linear dependencies among its rows. Let **G** denote a matrix the rows of which express these linear dependencies. That is, **G** satisfies the matrix equation:

$$\mathbf{G}\,\mathbf{N} = \mathbf{0}. \tag{11}$$

In mathematical terms, the rows of **G** span the left-hand side nullspace of **N** (i.e. the nullspace of the transpose of **N**). The nullspace of a matrix is the space of all vectors that give the null vector when multiplied by that matrix. Standard methods exist [72] and are usually included in computer mathematics packages, for computing the nullspace of a matrix such as **N**. Multiplying Eq. (4) by **G** from the left yields:

$$\mathbf{G}\frac{\mathrm{d}\mathbf{S}}{\mathrm{d}t} = \mathbf{0}. \tag{12}$$

Integration over time gives:

$$\mathbf{G}\,\mathbf{S}(t) = \mathbf{G}\,\mathbf{S}(0). \tag{13}$$

$\mathbf{S}(0)$ is the concentration vector at time zero and, thus, a vector of constants.

For the enzyme reaction (9), matrix **G** can be chosen to be:

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \tag{14}$$

which, given that the columns of **G** correspond to the rows of **N**, yields the conservation relations $E + ES = \text{const.}$ and $S + ES + P = \text{const.}$ The second relationship expresses the fact that the system is modeled as closed, i.e. both S and P are considered to be internal, and there is no mass flow into or out of the system via external metabolites.

The search for conservation relations is an example of a structural analysis, which does not require the knowledge of kinetic parameters such as Michaelis–Menten constants. It is helpful because, if the system involves conservation quantities, it allows one to reduce the dimension of the system equation (4). In fact, only a set of linearly independent differential equations needs to be considered since any dependent concentrations can be calculated from Eq. (13). Each conservation relationship allows for the removal of one dependent variable, though whether it is *E* or *ES* that is considered to be dependent in the above example is a matter of choice. [In principle it should not make any difference which is chosen, but there may be problems of numerical accuracy if the solution of Eq. (13) involves small differences between large concentrations.] Knowledge of conservation relations is also of importance in stability analysis [51] and for the quasi-equilibrium approximation (see Section 4.1).

Most conservation relations reflect the conservation of chemical entities such as the phosphate group and can thus be chosen to involve non-negative coefficients only. A method for calculating the coefficients in this case was given [117]. However, conservation relations may occur due to other reasons. These may not have this non-negativity property [117] or may even be nonlinear [110]. Biological implications of conservation relations have been discussed in Ref. [29].

3.2 Stationary States and Stability Analysis

Many synthesis or degradation processes in metabolism subsist (approximately) at a stationary state. This is not to be confused with a thermodynamic equilibrium state, in which all net reaction rates are zero and no entropy is produced. Cellular metabolism is an open system, so that there is a permanent throughput of mass. Very often, the input and output are balanced in such a way that the concentrations of metabolites do not change. This is in fact due to stability properties because a steady state can only be observed when it is stable, i.e. when small fluctuations are damped. Examples of stable steady states from everyday life are provided by a calmly flowing river and a light bulb converting electricity into light. Examples of unstable steady states are provided by a vibrating violin string and waves induced by the wind blowing over the ocean.

Examples of stable steady states in metabolism are glycolysis (degradation of glucose to give pyruvate or lactate and concomitant ATP production, see Figure 1) in many cells, amino acid synthesis and, at a lower level, the turnover of the enzyme–substrate complex in almost any enzymatic reaction.

At steady state, Eq. (4) simplifies to the algebraic equation system:

$$\mathbf{N}\,\mathbf{v} = \mathbf{0}. \tag{15}$$

For example, for the glucose phosphorylation system (1) in Section 2.2 with the stoichiometry matrix (5), this reads $v_1 - v_2 = 0$, which correctly describes the steady state for this system if only glucose-6-phosphate is considered to be an internal metabolite.

A large part of the theory underlying the modeling of metabolic networks is based on the steady-state equation (15). This equation is very attractive due to its simplicity. Moreover, it has a rather wide range of applicability because so many biochemical systems subsist in steady state. In the case of small fluctuations, it can still be used as an approximation; even when the system behaves in an oscillatory manner with constant amplitude, the averaged fluxes obey that equation because the internal metabolites must not accumulate nor run down in the long term. The difference between the terms *reaction rate* and *flux* is worth mentioning. The former term is used for reaction velocities at any state, e.g. at any time point of an oscillation, whereas the latter term is normally used for a reaction rate at a steady state.

To check whether a steady state is stable, stability analysis can be applied. This analysis is, strictly speaking, beyond structural (network) analysis because it requires some knowledge of kinetic parameters. Nevertheless, we treat it in this section because it is closely related to the study of steady states. A stability analysis is called local if the idealized situation of infinitesimally small fluctuations is considered.

To explain the basic idea of local stability analysis, we consider the simple case of two sequential monomolecular reactions, A → S → B, with S being considered to be an internal metabolite. Thus, we deal with a system that is one-dimensional (1-D) in terms of concentrations. Let the first reaction have a constant rate, v_{in} (input), while the second reaction follows the Michaelis–Menten kinetics given in Eq. (6). The steady-state equation $v_1 - v_2 = 0$ then reads:

$$v_{\mathrm{in}} = \frac{V_{\mathrm{max}} S}{K_{\mathrm{m}} + S} = 0\,, \tag{16}$$

Solving this for S gives:

$$S = \frac{v_{\mathrm{in}} K_{\mathrm{m}}}{V_{\mathrm{max}} - v_{\mathrm{in}}}\,. \tag{17}$$

This term is positive if $V_{\mathrm{max}} > v_{\mathrm{in}}$. In the opposite case, the limiting rate of the second enzyme is not sufficient to convert S with the same rate as it is fed into the system. Checking stability in this system can be done in a graphical manner, by plotting $\mathrm{d}S/\mathrm{d}t$ ($\dot{S}$ for short) versus S (Figure 3). The intersection of this curve with the S-axis gives the steady state. Assume there is a small fluctuation of the concentration, S, decreasing it. This leads to a positive value of $\dot{S}$, such that S increases. In contrast, a positive deviation leads to a negative

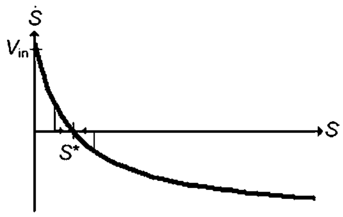


Figure 3 Plot illustrating the stability analysis of a simple system consisting of two sequential monomolecular reactions. S^* denotes the steady-state concentration. A negative deviation leads to a positive value of $\dot{S}$, such that S increases, while a positive deviation leads to the opposite effect. In either case, the system is running back to the steady state, so that this state is stable.

value of $\dot{S}$, such that S decreases. In either case, the system is running back to the steady state and, hence, this state is stable. For any 1-D system, a steady state is stable if the curve $\dot{S}$ versus S intercepts the S-axis with a negative slope.

In systems of dimension greater than one, stability analysis is more complicated. It is usually performed by using the so-called Jacobian matrix. As this is beyond the scope of this chapter, the reader is referred to Refs. [51,70].

3.3 Constraints on Steady-state Fluxes

If the fluxes in the steady-state equation (15) are regarded as variables rather than functions, then the equation is usually underdetermined, meaning that there is no unique solution for them. However, the steady-state requirement places constraints on their values, e.g. in a simple linear pathway of single-substrate, single-product reactions, the steady state would require that the fluxes were all identical, without specifying what their value would be. Just as the conservation relations between metabolites correspond to the left-hand side nullspace of matrix $\mathbf{N}$, it turns out that the relationships between fluxes are given by the right-hand side nullspace. This can be represented by the column vectors of a matrix $\mathbf{K}$ with maximum rank fulfilling an equation similar to Eq. (15) (e.g. Ref. [72]):

$$\mathbf{N}\,\mathbf{K} = \mathbf{0}\,. \tag{18}$$

Thus, the nullspace to the stoichiometry matrix is the region of all flux vectors that are, in principle, possible at steady state.

For identifying metabolic pathways, it is of interest to find very simple flux distributions. As an example, consider the simple network in Figure 4(A) consisting of a short linear pathway with a substrate cycle (e.g. Ref. [31]). The cycle would have to be driven by, for example, ATP hydrolysis, but ATP and ADP are not shown because they are being considered external metabolites in this case. Eq. (15) for this system is:

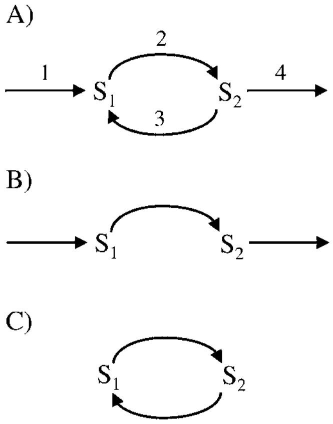


Figure 4 (A) Simple reaction scheme involving a substrate cycle. External metabolites are not shown. Pathways corresponding to the nullspace vectors (1 1 0 1)^T and (0 1 1 0)^T are depicted in (B) and (C), respectively.

$$\begin{pmatrix} 1 & -1 & 1 & 0 \\ 0 & 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}. \tag{19}$$

A possible basis for the nullspace of the stoichiometry matrix is:

$$\mathbf{K} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}, \tag{20}$$

where the two vectors correspond to two component pathways shown in Figure 4(B and C). The vectors do not contain the actual fluxes, but any observable flux vector can be expressed as a scaled combination of these two vectors, i.e.:

$$\begin{pmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \lambda_1 \\ \lambda_1 + \lambda_2 \\ \lambda_2 \\ \lambda_1 \end{pmatrix}, \tag{21}$$

where λ_1 and λ_2 are scaling factors. Apart from substrate cycles [31], biochemical pathways of amino acid synthesis, for example, have been analyzed using the nullspace approach [131, 136].

For many biochemical reactions, it is known whether they are reversible or irreversible. As mentioned in Section 2.4, in metabolic network analysis, irreversibility is not meant to exclude that the reaction involves a reverse step; rather, the reverse step is supposed to always have a lower rate than (or at most the same rate as) the forward step. Thus, the net flux is always non-negative for irreversible reactions. For example, most phosphatases (EC numbers 3.1.3.x) are irreversible, while all isomerase reactions (EC 5.x.y.z) are reversible. Information on the directionality of reactions is available much more often than kinetic data. It can be obtained from the original biochemical literature, textbooks and databases such as the authoritative NIST database of thermodynamics of enzyme-catalyzed reactions (http://xpdb.nist.gov/enzyme_thermodynamics/enzyme_thermodynamics_data.html) or somewhat indirectly and not always correctly from KEGG (http://www.genome.ad.jp/kegg) and BRENDA (http://brenda.uni-koeln.de). For example, the hexokinase reaction mentioned in Section 2.2 has such a high equilibrium constant that it is usually written as an irreversible reaction:

glucose + ATP → glucose-6-phosphate + ADP.

Without loss of generality, the sign definition of reaction rates can be chosen in such a way that the subvector in **v** corresponding to the irreversible reactions satisfies the inequality:

$$\mathbf{v}^{\mathrm{irr}} \geq \mathbf{0}\,. \tag{22}$$

The admissible region determined by relations (15) and (22) in flux space is a convex polyhedral cone (Figure 5). For instance, $[-1\ -1\ 0\ -1]^{\mathrm{T}}$ is a mathematically valid nullspace vector for our example substrate cycle (Figure 4), but uses the three reactions in the "wrong" direction and so is physically invalid. Of course, in this case the problem can be easily fixed by multiplying the vector by -1, thus reversing all the reactions. However, $[1\ 0\ -1\ 1]^{\mathrm{T}}$

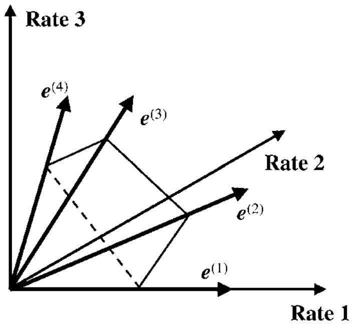


Figure 5 Schematic representation of a flux cone. The coordinates are the reaction rates of enzymes. The flux cone shown here is 3-D. Usually, the dimension of the space of rates (in which the cone is embedded) is then higher than three; only three axes are shown. Each generating vector (edge), $\boldsymbol{e}^{(i)}$, of the cone represents an elementary mode. Note that here the number of generating vectors is larger than the cone's dimension. The thin lines indicate a normalization by Eq. (27) and help to visualize the shape of the cone.

is also a valid nullspace vector, but this cannot be simply resolved, since reversing it merely changes which reactions are being used in the wrong direction. Moreover, a full set of linearly independent basis vectors often does not involve all biochemically meaningful pathways.

3.4 Defining Component Pathways of a Network

Based on early attempts in chemistry [17], methods for defining biochemical pathways in a formal way were developed [79, 116, 129]. Note that it is unnecessary to split the reversible reactions into forward and reverse steps, although this is sometimes done. Relations (15) and (22) constitute a linear equation/inequality system where, as in the example above, the solutions $\mathbf{v}$ of this system are indeterminate with respect to scaling. Thus, it is sensible to consider relative flux distributions, called flux modes.

A *flux mode*, $\boldsymbol{M}$, is defined as the set:

$$\boldsymbol{M} = \{\mathbf{v} \in R^r | \mathbf{v} = \lambda \mathbf{v}^*, \lambda > 0\} \,, \tag{23}$$

where r and λ are the number of reactions and an arbitrary real positive number, respectively, R^r denotes the r-dimensional Euclidean space, and $\mathbf{v}^*$ stands for a vector (different from the null vector) fulfilling the following two conditions:

(C1) Steady-state condition: $\mathbf{v}^*$ fulfills Eq. (15).

(C2) Sign restriction for irreversible reactions: if the system involves irreversible reactions, then the corresponding subvector $\mathbf{v}^{\text{irr}}$ of $\mathbf{v}^*$ satisfies inequality (22).

According to this definition, a flux mode can be characterized by one representative vector. A transformation route in metabolism can be formalized as a simple flux vector satisfying Eq. (15) and inequality (22) because along such a route, all intermediates must be balanced with respect to production and consumption and the irreversible enzymes must operate in the appropriate direction. The rationale for looking for a *simple* flux distribution rather than an arbitrary distribution fulfilling relations (15) and (22) arises from the general scientific paradigm of decomposing a system under study into its simplest components. Examples of this paradigm are Fourier analysis or the decomposition of gene expression patterns into principal components [1]. Decomposition under the side constraint of non-negativity plays a role also in pattern recognition, where the method of non-negative matrix factorization is used [73]. Dealing with simple routes (i.e. enzyme sets involving only a few enzymes) is particularly interesting in functional genomics, because the set of identified metabolic genes is often incomplete, so that it is meaningful to test whether at least one functional route can be realized by this set [20].

The crux of a proper pathway definition is to define what we mean by simplicity of the flux vector. An appropriate definition is the following: a flux mode, $\mathbf{v}^*$, is called elementary if there is no other flux distribution involving only a subset of the enzymes involved in $\mathbf{v}^*$. More exactly, this can be written in the following way [121]: a flux mode $\boldsymbol{M}$ with a representative $\mathbf{v}^*$ is called an *elementary flux mode* if, and only if, $\mathbf{v}^*$ fulfils the condition:

(C3) Simplicity (nondecomposability). There is no vector $\mathbf{v}'$ (different from the null vector) with the following properties: (i) $\mathbf{v}'$ obeys restrictions (C1) and (C2), and (ii) $\mathbf{v}'$ contains zero components wherever $\mathbf{v}^*$ does and includes at least one additional zero component.

Note that, upon testing whether a vector $\mathbf{v}^*$ is elementary, the test vector $\mathbf{v}'$ need not have the same values of the nonzero components as $\mathbf{v}^*$. Elementary modes can be visualized as follows. It can be shown that if all reactions are irreversible, each elementary mode corresponds to one edge of the cone (see Figure 5) determined by relations (15) and (22) and *vice versa* [116]. If some reactions in the system are reversible, there may be additional elementary modes in the interior of the cone [97].

In the case where all reactions are irreversible, the elementary modes can be computed by enumerating the edges of the cone. This enumeration can be performed by tableau methods starting from the stoichiometry matrix augmented with the identity matrix [88] or from the nullspace matrix [149]. These algorithms have been adapted to the case where some elementary modes lie inside the cone [120, 121, 144]. The extended algorithms have been implemented in METATOOL (http://pinguin.biologie.uni-jena.de/bioinformatik/networks) [97, 146] and several of the simulation packages mentioned in Section 4.1.

Whereas the choice of nullspace vectors to form a basis, as previously described in Section 2, was not unique, the elementary modes are uniquely determined (up to multiplication by a non-negative factor). Moreover, each flux mode possible in the living cell can be written alternatively as a linear combination of basis vectors or of elementary flux modes. If the combination in terms of elementary modes is used, all coefficients, α_k, must be non-negative [121]:

$$\mathbf{v} = \sum_k \alpha_k \mathbf{e}^{(k)} \,, \quad \alpha_k \geq 0 \,, \tag{24}$$

where $\mathbf{e}^{(k)}$ denotes vectors representing elementary modes. This non-negativity condition needs to be included because the sign restriction for irreversible reactions must be satisfied. Importantly, enzyme deficiencies and gene knockouts can properly be described by elementary-mode analysis, but not with the nullspace approach. Upon deletion of an enzyme, all elementary modes not involving this enzyme remain unchanged, while the basis vectors of the nullspace need to be recalculated in some cases. Consider, for example, the system depicted in Figure 4. Deletion of enzyme 2 in that system would imply that both nullspace vectors given in Eq. (20) drop out, although the remaining system still has a 1-D nullspace [represented by the vector $(1\ 0\ -1\ 1)^{\mathrm{T}}$, which may be relevant if reaction 3 is reversible].

One inconvenience of the larger number of elementary modes than nullspace vectors is that there may be multiple solutions for the values of the α_k in equation (24) that relate the elementary modes to an observed flux vector [152]. However, a least-squares solution (which minimizes the norm of the α vector) is uniquely defined and has advantages as a practical solution that facilitates the interpretation of changes in the utilization of different elementary modes by an organism as environmental conditions change [102].

3.5 Examples of Elementary-modes Analysis

The elementary modes of a list of reactions of carbohydrate metabolism include familiar metabolic pathways such as glycolysis, and various functionalities of the oxidative and nonoxidative pentose phosphate pathways (Figure 1), as well as other pathways catabolizing glucose that are not generally named [120]. It is worth noting that, although the term "pentose phosphate pathway" is firmly established in biochemistry, there is not just one pathway deserving this name. Rather, it occurs in several functionalities [138], as confirmed by elementary-modes analysis [120]. This demonstrates that elementary modes have every right to be regarded as biochemical pathways. In this section, we will show the "added value" of elementary modes in avoiding incorrect interpretations and in deducing novel interpretations of metabolism.

Several authors have used graph-theoretical concepts to define metabolic pathways [60,76,128]. However, paths traced on graphs may not be competent metabolic pathways. An illustration of this point recapitulates a controversy from the history of biochemistry. Figure 6 shows the pathways of the tricarboxylic acid (TCA) cycle and gluconeogenesis to glucose. It has long been considered that, given an input of acetyl-CoA from the breakdown of fatty acids or ketogenic amino acids, it is not possible for animals to achieve net synthesis of glucose from this precursor. However, it is evidently possible to trace a route from acetyl-CoA, around the TCA cycle to oxaloacetate, from oxaloacetate to phosphoenol pyruvate (PEP) and from there to glucose (Figure 6A). Indeed, with the introduction of ^{14}C isotopic labeling in the 1950s, it was established that isotope could pass along this apparent pathway. Nevertheless, animals cannot make glucose from two-carbon precursors in a sustained steady state. All organisms that can (e.g. green plants and many bacteria) contain the glyoxylate cycle, yielding the metabolic network illustrated in Figure 6(B). The potential flows of carbon in this network can be investigated by regarding all cosubstrates (ATP, NAD, etc.) as external and computing the elementary modes (e.g. by METATOOL). This results in two modes that completely oxidize acetyl-CoA to CO_2 and the only mode that generates glucose from acetyl-CoA does so via the glyoxylate cycle enzymes. The reason is that the TCA cycle alone cannot cause a net synthesis of oxaloacetate; for every acetyl-CoA that enters the cycle, two carbons are lost as CO_2, so none is available to form glucose.

This example illustrates that if only the connectedness of the graph is considered and the stoichiometric constraints are neglected, then it is likely that nonfunctional pathways will be postulated. Moreover, this is not an isolated case: significant problems about our understanding of plant plastid metabolism have been revealed by elementary-modes analysis [100, 101].

Figure 6(C) indicates how elementary modes can identify previously unrecognized pathways. The elementary modes computed for the network include two different modes for complete oxidation of glucose to CO_2. One of the modes corresponds to oxidation via the TCA cycle as normally described in textbooks. The alternative pathway uses the glyoxylate cycle in catabolic mode; the oxaloacetate produced by the glyoxylate cycle is converted to PEP, with loss of one CO_2, and then to pyruvate before decarboxylation to acetyl-CoA with loss of a second CO_2. The oxaloacetate produced by malate dehydrogenase is used in equal proportions by the enzymes leading to citrate and PEP. The ATP yield of this pathway per mole of glucose is smaller than that of the usual TCA cycle (1 instead of 2). We originally noted this pathway [119], which is part of a larger pathway predicted for *Escherichia coli* earlier [74]. This theoretical prediction has been confirmed experimentally, as the occurrence of this pathway in *E. coli* growing slowly at low glucose concentrations has recently been shown by Fischer and Sauer [34] (preliminary observations have been presented in Ref. [153]). Since the enzymes involved in the pathway are present in many microorganisms, it can be supposed that it operates also in other microbes.

An example of how elementary-modes analysis can shed new light on plant metabolism is provided by a special form of photosynthesis. Crassulacean acid metabolism (CAM) is a variant of photosynthesis employed by a range of plants (e.g. cacti) as an adaptation to arid conditions [22, 141]. In order to reduce water loss, the stomata are closed during the daylight hours, so CO_2 cannot be obtained directly from the atmosphere. Instead, the plants fix CO_2 during the night by breaking down stored carbohydrate to give PEP, which is then carboxylated to oxaloacetate and reduced to malate, which is stored in the vacuoles. In the daytime, malate is released from the vacuoles and decarboxylated in the cytoplasm; the CO_2 released then diffuses into the chloroplast where it is fixed by ribulose bisphosphate carboxylase (rubisco) and the rest of the Calvin cycle. CAM plants vary with respect to the enzyme

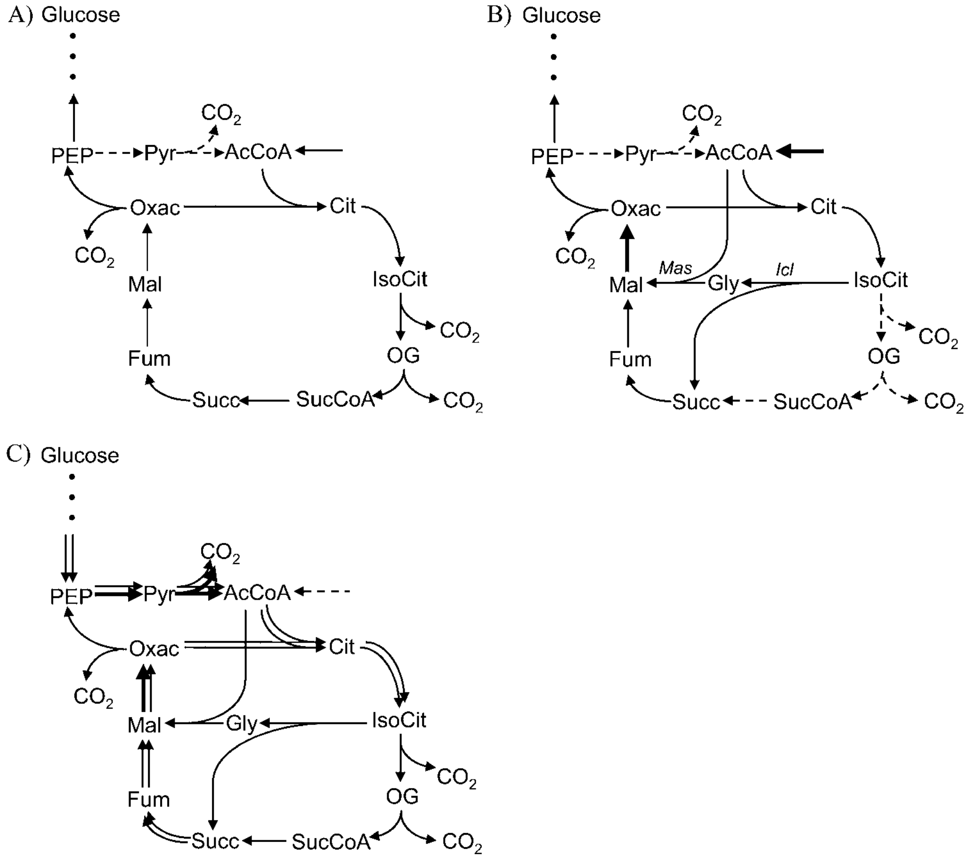


Figure 6 Scheme of the TCA cycle and gluconeogenesis. Thick arrows represent fluxes that are double as high as through the other reactions. Dashed arrows represent unused enzymes. Abbreviations of metabolites: AcCoA, acetyl-coenzyme A; Cit, citrate; Fum, fumarate; Gly, glyoxylate; IsoCit, isocitrate; OG, oxoglutarate; Oxac, oxaloacetate; PEP, phosphoenolpyruvate; Mal, malate; Pyr, pyruvate; Succ, succinate; SucCoA, succinyl coenzyme A. (A) Situation in mammals. A pathway from acetyl-CoA to glucose seems to exist (solid arrows). However, Oxac is not stoichiometrically balanced at steady state. (B) Situation in plants and many bacteria, where the TCA cycle involves the glyoxylate shunt, consisting of isocitrate lyase (*Icl*) and malate synthase (*Mas*). This allows conversion of acetyl-CoA into glucose at steady state (pathway in blue). (C) Classical TCA cycle (red); pathway of glucose catabolism alternative to the classical TCA cycle (green).

used to break down the malate, which is either PEP carboxykinase (PEPCK) or malic enzyme [23]. They also vary in whether the primary product of photosynthesis is chloroplastic starch or cytosolic hexoses. On this basis, Christopher and Holtum [16] proposed there were potentially four distinct groups of CAM plants described by these two criteria, and they classified a range of plants to show that each of the four categories had members.

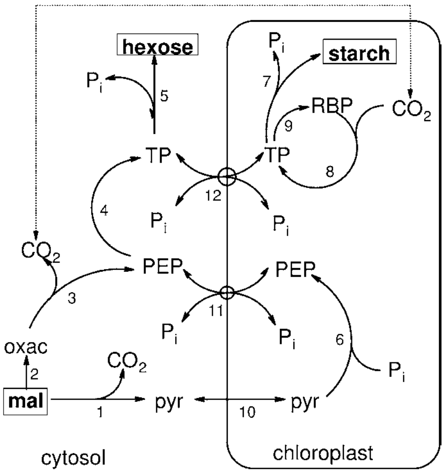


Figure 7 Reaction scheme of CO_2 fixation in CAM plants. Reactions: 1, malic enzyme; 2, malate dehydrogenase; 3, PEPCK; 4, enolase and phosphoglyceromutase; 5, gluconeogenesis; 6, pyruvate, phosphate dikinase; 7, starch synthesis; 8, rubisco; 9, Calvin cycle; 10, pyruvate translocater; 11, PEP translocator; 12, phosphate/triose phosphate translocator.

This is an interesting problem to examine with elementary modes, since we already knew that the metabolic exchanges between chloroplast and cytosol allowed (or not) by the chloroplast membrane metabolite transporters were a significant constraint on feasible plant metabolism [100]. A scheme of the metabolism under study is shown in Figure 7 and the stoichiometries of the reactions are given in Table 1. The model has been drawn up to represent carbon and phosphate balancing in the reactions, but redox balance is not represented. Figure 8 shows the six elementary modes of this scheme that convert malate to hexose and/or starch. There are three modes each using malic enzyme and PEPCK, respectively, to release CO_2 from malate. For each of the modes leading to one product only, Christopher and Holtum [16] gave examples (Figure 8). For example, in pineapple (*Ananus comosus*), sugar is produced via PEPCK. The reason that, in addition to the modes producing hexose only and starch only, there is a third mode coproducing hexose and starch is that the latter involves no net flux through the triose phosphate translocator and is, hence, no superposition of the former two. In the coproduction modes, the carbon in the hexose comes from the triose, whereas the starch comes from the CO_2 derived from malate breakdown; the hexose to starch stoichiometry is 3:1.

The question then arises whether coproduction of starch and hexose defines a subset of CAM plants. In fact, Christopher and Holtum [16] noted that the PEPCK-containing plant *Aloe vera* produced both hexoses and starch during the day, in the ratio of about 2:1 from their figures, although they

Table 1 Reactions of the CAM plant metabolism model

No.	Step	Reaction
1	ME	mal + NAD → pyr_cyt + NADH + CO_2
2	MDH	mal + NAD → oxac + NADH
3	PEPCK	oxac + ATP_cyt → PEP_cyt + ADP_cyt + CO_2
4	glc1	PEP_cyt ↔ tp_cyt
5	glc2	2tp_cyt → hex_cyt + 2Pi_cyt
6	PPDK	pyr_chl + P_i_chl + ATP_chl → PEP_chl + PPi + AMP_chl
7	StSyn	2tp_chl → starch + 2P_i_chl
8	RbCo	RBP + CO_2 → 2tp_chl
9	CC	5tp_chl + P_i_chl → 3RBP
10	PyrTr	pyr_cyt ↔ pyr_chl
11	PEPT	P_i_cyt + PEP_chl ↔ P_i_chl + PEP_cyt
12	TPT	tp_cyt + P_i_chl ↔ tp_chl + P_i_cyt
13	OP	ADP_cyt + P_i_cyt → ATP_cyt
14	AK	AMP_chl + ATP_chl ↔ 2ADP_chl
15	LR	ADP_chl + P_i_chl → ATP_chl
16	PPase	PP_i → 2P_i_chl

The full names for steps 1–12 are given in the legend to Figure 7. Malic enzyme is shown as the NAD version, but since nicotinamide nucleotides (NADP, etc.) are treated as external in the model (and are not shown in the grouped reactions), the NADP-linked enzyme is implied as well. The reactions 13–16 not shown in Figure 7 are added to represent recycling of adenine nucleotides and pyrophosphate in the appropriate compartments: OP, oxidative phosphorylation; AK, chloroplast adenylate kinase; LR, light reactions, i.e. photophosphorylation; PPase chloroplast pyrophosphatase. Irreversible reactions are designated by "→" and reversible ones by "↔". Except for CO_2, metabolites occurring both in cytosol and chloroplast are represented twice with designations "cyt" and "chl", respectively, and are treated as separate entities for modeling purposes. Other abbreviations are: mal, malate; pyr, pyruvate; oxac, oxaloacetate; tp, triose phosphate; RBP, ribulose 1, 5-bisphosphate; hex, hexose sugars; PP_i, pyrophosphate.

classed it as a hexose producer. This leaves hexose and starch-producing malic enzyme plants as the category to be filled. Christopher and Holtum characterized CAM-induced *Mesembryanthemum crystallinum* as a starch-producing malic enzyme plant. However, it seems that it can produce both hexose and starch [6]. Furthermore, even when starch is a major product, study of diurnal variation in the transporter transcripts and activities [49] shows that the triosephosphate transporter activity does not increase in the day, whereas the glucose-6-phosphate transporter does, suggesting that cytosolic hexose rather than triose phosphate is used as a substrate for starch synthesis. This is more like the pattern in Figure 8(C) than Figure 8(B). A better example of a malic enzyme plant producing both starch and hexose, however, is *Clusia minor* [7]. The experiments reported a ratio of formation of hexose to glucans and starch of 2:1; although this is lower than the hexose to starch ratio of the elementary mode, the glucan component is likely to be cytosolic. In all, there is an arguable case that there are six modes of operation of CAM, as revealed by elementary modes, with plants that predominantly use each of

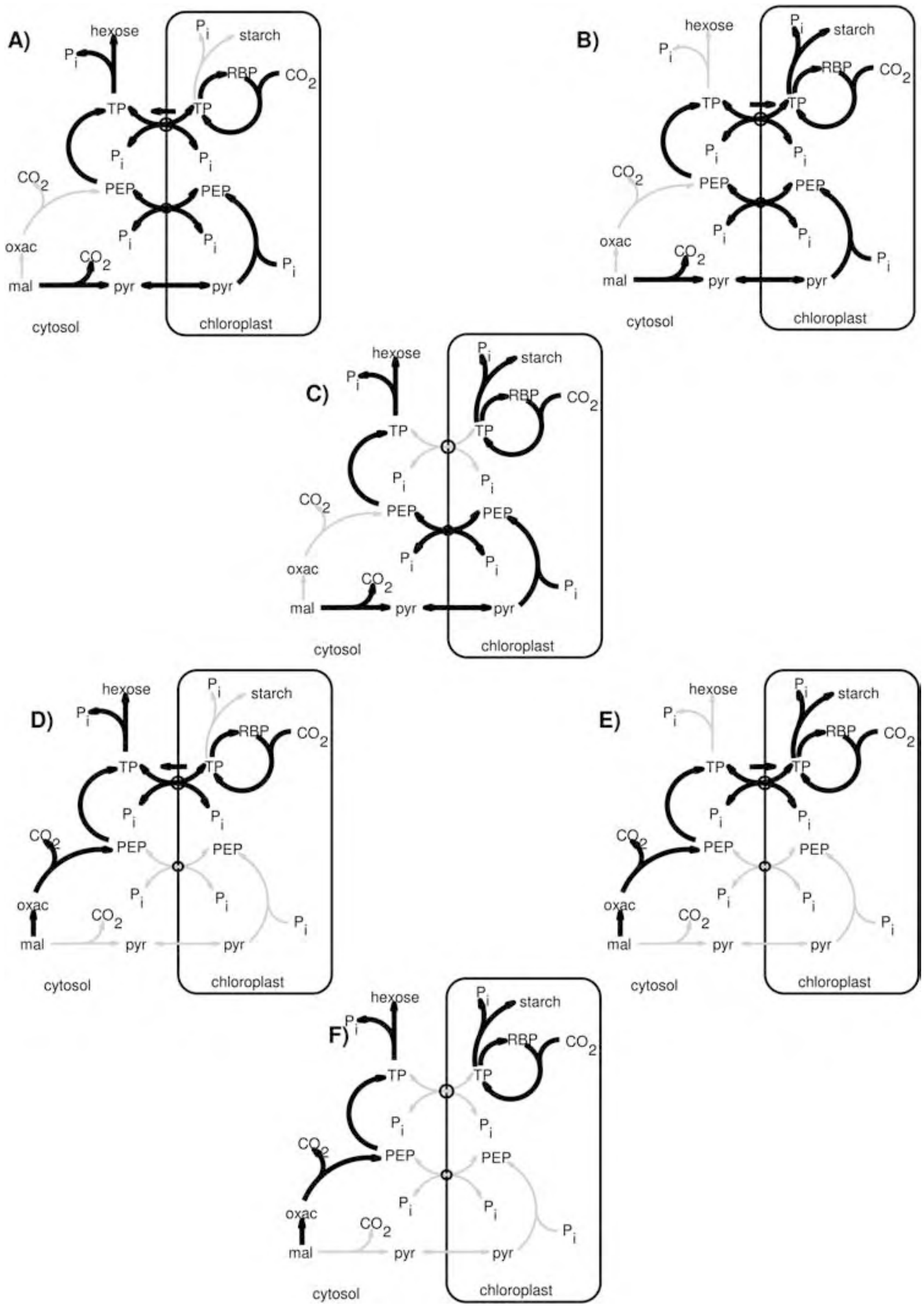

A)
hexose
P_i
starch
RBP
CO_2
TP
TP
P_i
P_i
CO_2
PEP
PEP
P_i
P_i
oxac
CO_2
P_i
mal
pyr
pyr
cytosol
chloroplast
B)
C)
D)
E)
F)

Figure 8 The six elementary modes occurring in the system shown in Figure 7. (A) Hexose synthesis via malic enzyme as occurring, for example, in Agavaceae and Dracaenacea. (B) Starch synthesis via malic enzyme as occurring, for example, in Cactaceae and Crassulacea. (C) Simultaneous starch and hexose synthesis via malic enzyme as occurring, for example, in *Clusia minor*. (D) Hexose synthesis via PEPCK as occurring, for example, in *Clusia rosea* and *Ananus comosus*. (E) Starch synthesis via PEPCK as occurring, for example, in Asclepadiaceae. (F) Simultaneous starch and hexose synthesis via PEPCK as occurring, for example, in *Aloe vera*.

them. Whereas, in general, metabolic flux distributions are a superposition of several elementary modes, the considered part of the metabolism in CAM plants is a case where virtually "pure" elementary modes occur. Another such case is glycolysis in mature yeast cells.

3.6 Extreme Pathways

In many applications, e.g. in the analysis of robustness [91, 103] (see Section 3.8), the concept of extreme pathways [108] is used. This differs only slightly from that of elementary modes. One of the goals is a reduction in the number of pathways to cope with the problem of combinatorial explosion of the number of pathways in large networks. As mentioned above, the admissible flux region determined by relations (15) and (22) is a convex cone (Figure 5), and there may be some elementary modes lying in the interior of the cone. It has been argued that for a minimal description of the system, the edges of the cone (sometimes called the *convex basis*) are sufficient; each admissible flux distribution can be written as a non-negative linear combination of the convex basis vectors [89, 97]. However, the elementary modes in the interior of the cone often represent biochemically meaningful pathways, such that it is not really justified to drop them.

Schilling and coworkers [108] have tackled this issue in the following way. They make a distinction between exchange reactions and internal reactions. The former connect to external metabolites, while the latter do not. Then, all reversible internal reactions are decomposed into forward and reverse steps. The convex basis vectors of the resulting flux cone are defined to be the set of extreme pathways. They have the advantage of a one-to-one correspondence to the edges of the flux cone. Note, however, that the respective flux cone is of higher dimension than the original one because the decomposition of reversible internal reactions adds dimensions to the flux space. In effect, many of the internal elementary modes are brought to the surface of the flux cone.

As shown in Ref. [67], elementary modes and extreme pathways are equivalent if all exchange reactions are irreversible. This is the case in several simulations [91, 103]. If there are reversible exchange reactions, the number of extreme pathways may be considerably smaller than the number of elementary modes [92], which is favorable in view of the problem of combinatorial explosion. However, care must be taken that no external metabolite takes part in more than one reversible exchange reaction. Otherwise, there would be a reversible extreme pathway, because that metabolite can be produced by one reaction and consumed by the other, and the opposite process is possible as well. This would be in contradiction to the desired property that extreme pathways are edges of a pointed cone. Therefore, for this case, Schilling and coworkers [108] suggested making the external metabolites connecting to these exchange reactions internal and extending the system by adding formal exchange fluxes connecting to these metabolites. They called this, together with the decomposition of reversible internal reactions, the reconfiguration of the system. This, however, generates an extended system in which extreme pathways and elementary modes are again equivalent [115]. For a comparison of the two concepts, see also Refs. [67, 92].

3.7 Optimization of Molar Yields and Flux Balance Analysis (FBA)

In biotechnology, one is often interested in increasing the molar yield of a given biotransformation, i.e. the molar ratio indicating how many molecules of product are formed per mole of substrate. Some product of interest may be synthesized on several routes with different yields, as in lysine synthesis [21] or tryptophan synthesis [119]. Thus, one may try to overexpress the enzymes along the route allowing maximum yield and possibly suppress other enzymes. A flux mode using only the optimal route is certainly not feasible alone since other products (e.g. other amino acids) must also be synthesized. Nevertheless, it is of interest to calculate upper bounds on the molar yield, to learn what can at best be expected, especially as many products of interest are formed in the stationary phase where there is little biosynthetic activity, and the cell's remaining requirements are largely for "maintenance energy".

The molar yield of a biochemical transformation can be expressed as:

$$\eta = \frac{\text{rate of product synthesis}}{\text{rate of substrate consumption}} . \tag{25}$$

Let S and P stand for the substrate and product of interest, respectively. [Note that we here allow for a larger number of reactions between S and P than in Eq. (9).] We denote their stoichiometric coefficients by m_{Sk} and m_{Pk}, respectively, with k being the reaction index. Then the optimization criterion can be written as:

$$\text{maximize } \eta = -\frac{\sum_k m_{Pk} v_k}{\sum_k m_{Sk} v_k} , \tag{26}$$

where the minus sign enters because the coefficients m_{Sk} are negative. Note that the objective function is nonlinear, with the variables being the fluxes v_k. As Eq. (26) is invariant to a scaling of the v_k by a common factor, and the side constraints (15) and (22) are linear and homogeneous, the solution to the maximization problem is indeterminate with respect to scaling. Thus, usually the substrate consumption rate is written in a normalized form:

$$-\sum_k m_{Sk} v_k = 1 \, . \tag{27}$$

Substituting this into Eq. (26) yields a linear objective function. Sometimes, Eq. (27) is phrased as an inequality imposing an upper bound on the input flux. Yet, in most cases, maximization of the numerator in Eq. (26) implies that the upper bound is attained by the system, such that an inequality side constraint is equivalent to Eq. (27).

Since any flux mode is a superposition of elementary modes with nonnegative coefficients (see Eq. 24), the molar yield of a flux distribution is a weighted average of the yields of the elementary modes involved:

$$\eta = \frac{\sum_k \alpha_k \eta_k}{\sum_k \alpha_k} \, , \tag{28}$$

where the α_k are the weighting coefficients as appearing in Eq. (24). Therefore, η cannot exceed the maximum value of all η_k. Consequently, the optimal flux distribution with respect to maximizing the molar yield always coincides with an elementary mode. When two or more elementary modes realize the same, maximum yield, any linear combination of these implies that maximum yield.

Elementary-modes analysis was successfully applied in Ref. [74] for predicting maximum yields in the synthesis of precursors for aromatic amino acids in *E. coli*. Carlson and coworkers [13] studied a metabolic pathway model of a *Saccharomyces cerevisiae* strain that had been engineered genetically to produce poly-β-hydroxybutyrate (PHB). Adding the natively absent ATP citrate-lyase to the network, the maximum theoretical PHB-to-carbon yield was increased from 0.67 to 0.83. Recently, Schwender and coworkers [123] conducted a combined experimental and theoretical study (using elementary modes) on oilseed rape, and found a previously undescribed metabolic route with high carbon efficiency.

As the objective function of maximizing molar yields and the side constraints [relations (15) and (22)] are linear, the maximization problem can alternatively be solved by linear programming [26, 27, 33, 93, 145, 150]. That method finds the pathway with the highest yield. In the event that there are multiple solutions giving the same yield, care has to be taken that all of them are detected [78] since linear programming returns a single solution by default. In contrast, elementary-modes analysis provides a set of candidate solutions, from which the best for a given substrate–product pair, and consecutively for different substrate–product pairs, can be selected easily. This allows for detecting not only (possibly multiple) optimal solutions, but also suboptimal, equally simple situations. These might be realized more readily (at least approximately) in biotechnological setups than the optimal situation. Accordingly, this analysis gives more output information than linear programming.

One method in the framework of stoichiometric network analysis is *FBA*. While *metabolic pathway analysis* is about computing the simplest flux distributions (elementary modes or extreme pathways), which need not necessarily occur in living cells in pure form, FBA is about computing optimal flux distributions, with a stronger focus on predicting real situations in living cells. The main idea is that the molar yield of some important product has been maximized during evolution. This is a principle of economy – cells that use the resource as efficiently as possible are assumed to have a selective advantage. This approach can be traced back to the linear programming approaches mentioned above [33, 93, 150]. Later, the method was refined, extended and applied to various systems of increasing complexity, and the term FBA was coined [24–27, 57, 145]. In most cases, biomass has been taken as the product of interest. The method is attractive because it allows the computation of flux distributions in living cells based on very little data: the reaction stoichiometries, information about irreversibility and the weighting coefficients m_{Pk} in the objective function (Eq. 26). However, the question arises whether maximization of output yield is really an outcome of evolution for all cells under all circumstances. If this were so, why does baker's yeast (*S. cerevisiae*) use the respiro-fermentation pathway (leading mainly to ethanol and implying an ATP per glucose yield of slightly more than 2) even under aerobic conditions if sufficient glucose is available? In order to maximize the ATP per glucose ratio it should use pure respiration, which allows a yield of more than 30. An answer was given from the viewpoint of evolution [98]. Fermentation allows a very high ATP production rate, although yield is low. Due to the high rate, yeast can grow fast and outcompete other species, withdrawing their nutrients. However, other yeast genera such as *Kluyveromyces* do use respiration [44]. It is hard to explain this diversity in sugar degradation among yeast species by a single optimization principle.

Maximization of yield and maximization of rate are not necessarily equivalent principles [2, 98, 147]. For instance, the bacterium *Holophaga foetida* growing on methoxylated aromatic compounds switches from a high-rate regime to a high-yield regime when the substrate level becomes very low [64]. The pathway active in hungry *E. coli* mentioned in Section 3.5, however, has a lower ATP yield than the usual TCA cycle. Other examples are provided by dimorphic fungi, i.e. fungi that switch between unicellular and multicellular stages. *Mucor racemosus*, for example, mainly relies on fermentation in the unicellular stage and on respiration in the multicellular stage [58]. Interestingly, it is the other way round in *Candida albicans* [71]. In any case, the main hypothesis underlying FBA cannot apply to both forms of dimorphic fungi. Recent experimental data on *Bacillus subtilis* [35], and theoretical and computational results [54] cast doubt on the principle of maximizing yield as well. Thus, it may be questioned whether maximization of (biomass, ATP or any other) yield is an over-riding goal in biology. Nevertheless, FBA is a helpful tool in many situations, such as adaptive evolution of *E. coli* after a change in substrate [57] or after a knockout [36].

3.8 Analyzing the Robustness of Metabolism

A striking feature of living cells is their homoeostasis, i.e. they are robust to external and internal perturbations within some range. For example, many knockout mutants of microorganisms are able to grow, some showing almost the same growth rate as the wild-type. This has been demonstrated by a systematic study on single-knockout mutants of virtually all genes in *S. cerevisiae* [42, 155]. Although an analogous systematic study on double mutants is not yet feasible due to the large number of combinations, first attempts in this direction have been made [130, 143]. As mentioned in Section 3.7, many cells harbor parallel and, thus, redundant metabolic pathways. For example, the pentose phosphate pathway circumvents the upper part of glycolysis (Figure 1). However, this bypass implies a loss in ATP production. Often, redundancy in metabolism cannot be seen as easily as in this example. Theoretical tools are needed [66, 83, 135, 154] to understand robustness in complex systems such as metabolic networks.

Elementary-modes analysis (see Section 3.4) is well suited for analyzing the structural robustness of metabolic networks because each elementary mode is nonredundant, with redundancy meaning that an enzyme could be deleted without interrupting the transformation of a substrate into a product. Therefore, redundancy can be quantified by the number of such modes [135]. The concept of extreme pathways has also been used for analyzing redundancy [91, 103]. Simulations for amino acid metabolism of *Haemophilus influenzae* showed that there was an average of about 40 extreme pathways corresponding to the same input/output regime (the exact number depending on conditions), when its metabolic network was used to produce a single amino acid [91]. Similar calculations for *Helicobacter pylori* gave the result that the synthesis of amino acids and ribonucleotides in this bacterium is less redundant than in *H. influenzae*, with only about two extreme pathways per input/output regime [103]. In these studies, the number of extreme pathways with the same overall stoichiometry (in terms of initial substrates and final products) is used as a measure of redundancy.

A somewhat different approach was suggested in Ref. [12]. The importance of each enzyme was assessed by the number of elementary modes in which it is involved, i.e. by the number of modes disrupted when the enzyme in question is deficient. Along these lines, it has been argued that robustness is not perfectly identical to redundancy [154]. In defining structural robustness, the intact system should be compared with a mutated system. To characterize the structural robustness against the knockout (deficiency) of one enzyme, E_i, the ratio between the number of elementary modes remaining after knockout, $z^{(i)}$, and the number in the wild-type network, z, can be used. This gives a normalized value between zero and unity. The extreme values are reached when no elementary mode is left and when all elementary modes remain. To quantify the global robustness of the network, the arithmetic mean of all these numbers can be taken:

$$R_1 = \frac{\sum_{i=1}^{r} z^{(i)}}{r \cdot z} \, . \tag{29}$$

Apart from this general measure of robustness, we defined measures for the robustness of the synthesis of specific products [154]. For a model of the metabolism of human erythrocytes proposed in Ref. [151], the quantity R_1 was computed to be 0.3834, while for various subsystems of the amino acid metabolism in *E. coli* (e.g. Ref. [135]) this measure was computed to be in the range 0.5112–0.5479. This is in agreement with the well-known fact that erythrocyte metabolism is very susceptible to enzyme deficiencies while the metabolism of *E. coli* is rather robust.

4 Dynamic Simulation

4.1 How is a Dynamic Model Constructed?

Before constructing a dynamic model, one should decide what type of model is needed. There is a basic distinction between deterministic and stochastic models. The latter is needed if the molecule numbers are so low that fluctuations play an essential role. Indeed the concentrations of many intermediates are rather low, often much lower than the resolution of nuclear magnetic resonance measurements, which is of the order of 10 μM *in vitro* and 100 μM *in vivo*. For example, in the analysis of calcium oscillations, stochastic simulation is sometimes used due to the low cytosolic concentration of calcium [28]. Another distinction is made between spatially homogeneous and heterogeneous models. Spatial gradients inside living cells may play a role due to the highly organized structure of cell organelles, the cytoskeleton and multienzyme complexes. Still another distinction can be made between models that are discrete or continuous in concentrations (with most stochastic models being discrete).

Most metabolic models, however, are deterministic, spatially homogeneous and continuous [5, 14, 15, 39, 51, 59, 61, 70, 84, 124, 156]. This is because, in that way, they are much simpler and because the numerical computational techniques are better developed, so simulation using them is more rapid and accurate.

The basis for deterministic, spatially homogeneous and continuous models is Eq. (4) together with the rate laws $\mathbf{v} = \mathbf{v}(\mathbf{S})$. This gives a system of n ordinary differential equations in the variables S_i. The differential equations are generally nonlinear because some, if not most or all of, the rate laws are nonlinear in concentrations. Therefore, solving the system equation (4) in time is not normally feasible by analytical methods, so that numerical methods are employed.

Many general purpose computer mathematics packages will solve sets of nonlinear differential equations given that the user has prepared a set of equations in the required format [corresponding to the system equation (4)]. However, this is a tedious and error-prone task that needs to be repeated whenever the model is altered, since changing a rate function means finding all its occurrences in the differential equations (one for each substrate and product involved in the reaction concerned) and changing them. Therefore many authors have devised computer software that generates and solves the appropriate differential equations from a set of reaction definitions and rate equations, using either a set of predefined rate equations or others defined by the user or both. Such software also usually checks for linear dependencies and conservation equations (as explained in Section 3.1) and reduces the set of equations to be solved appropriately. Two different types of solution to the differential equations can be computed: integration of the equations to produce time courses of metabolite concentrations and fluxes, and solution of the equations for the substrate concentrations (and hence fluxes) giving a steady state [when all derivatives in system equation (4) equal zero]. Techniques for the former are much more dependable than those for the latter, which usually work by successive approximation from an initial estimate of the solution and are prone to failure if the initial estimates are not good enough. One way of generating appropriate initial estimates to improve the success rate is to simulate a time course as far as a close approach to a steady state and then to pass the substrate concentrations to the steady-state solver. It is unreliable to compute a time-course to an apparent steady state alone, since even when the differential equations are all near-zero, it is impossible to tell how far away the steady state is if there are slow processes in the system being studied.

In developing special software packages for both simulation and steady-state solution of metabolic systems, two different approaches have been adopted (e.g. Ref. [94]). GEPASI (http://www.gepasi.org) [80, 81] and DB-Solve [46] are examples of packages that are operated through a graphical user interface. The other approach is the command-line driven program, originally operating on pre-written scripts, but more recently working interactively. Examples include SCAMP [105], Jarnac [106], ScrumPy [100,102] and PySCeS [90] (see Table 2). The distinction is breaking down as more of the packages can inter-operate through the Systems Biology Workbench (SBW; http://sbw.kgi.edu) [106] and by model exchange through a common model definition language, Systems Biology Markup Language (SBML) [56]. They not only serve for numerical integration of differential equation systems, but also for computing steady states and checking their stability, and computing elementary modes, control coefficients (e.g. Refs. [32, 51, 70]) and other quantities. Table 2 gives an overview of widely used software packages for simulating metabolic networks. Many more packages have been developed, such as PLAS (http://www.dqb.fc.ul.pt/docentes/aferreira/plas), DBSolve [46] and the JJJ web simulator (http://jjj.biochem.sun.ac.za).

Table 2 Software for simulating metabolic networks

Software package	Analyses performed	Reference	URL
FluxAnalyzer	MFA, MPA	68	http://www.mpi-magdeburg.mpg.de/-projects/fluxanalyzer
METATOOL	MPA	97, 146	http://pinguin.biologie.uni-jena.de/-bioinformatik/networks/index.html
GEPASI/COPASI	CSS, DS, MCA, MPA, optimization, PE, in COPASI also stochastic simulation	80, 81	http://www.gepasi.org
JARNAC	CSS, DS, MCA, MPA	106	http://www.sys-bio.org
SBW (Systems Biology Workbench)	CSS, DS, MCA, MPA	106	http://sbw.kgi.edu
ScrumPy	CSS, DS, MCA, MPA, optimization, PE	100, 102	http://mudshark.brookes.ac.uk/ScrumPy
PySCeS	bifurcation analysis, CSS, DS MCA, MPA	90	http://pysces.sourceforge.net
ProMoT/Diva	DS, algebro-differential equations	43	http://www.mpi-magdeburg.mpg.de/en/research/projects/-1002/comp_bio/promot
E-Cell	DS, stochastic simulation	139	http://www.e-cell.org/software/ecellsystem

CSS, calculation of steady states; DS, dynamic simulation; LSA, local stability analysis; MCA, metabolic control analysis; MFA, metabolic flux analysis; MPA, metabolic pathway analysis; PE, parameter estimation.

When choosing the appropriate method for numerical integration, one should keep in mind that most biological systems show the phenomenon of timescale separation (time hierarchy). They involve slow and fast processes. Metabolic systems, for instance, are characterized by the presence of slow and fast enzymes. Isomerases, for example, are usually present in high concentrations so that they catalyze the corresponding reactions very fast. Time hierarchy makes biological systems simpler with respect to their dynamics. Nonlinear systems with more than two dimensions can potentially exhibit behaviors more complex than steady states or regular oscillations. For example, they may allow deterministic chaotic behavior, in which the system never returns to the initial state exactly and in which small deviations in the initial conditions lead to large deviations later [137]. However, most biological systems at the cellular level do not exhibit chaos. This is probably due to a reduction in the effective dimension by time hierarchy [50]. In mathematical terms, the differential equations describing systems showing time hierarchy are called *stiff*. Special numerical integration routines have been developed for dealing with stiff systems, such as the Gear procedure [41], but the development that brought metabolic modeling easily within the scope of the desktop computer was the routine LSODA [95], which was incorporated into GEPASI, SCAMP and ScrumPy amongst others. This is an adaptive routine that only switches into the more computationally demanding Gear-like algorithm when the solution is judged to be stiff. More recently, a successor to LSODA, CVODE [18], is being exploited in, for example, Jarnac.

The time hierarchy in biochemical systems can be usefully exploited in modeling. Any modeling is done for a certain timescale of interest. All processes slower than this timescale can be considered to be constant. As for the processes faster than this timescale, a simplifying method can be applied that is based on the reasoning that these processes have relaxed to a quasi-steady state after a very short transient time. There are two main types of simplification: the quasi-steady-state and quasi-equilibrium approximations (e.g. Refs. [19, 51, 70]). These two types are sometimes confused in the literature. In the former method, the assumption is made that some metabolite concentrations reach a quasi-steady state while others do not. That is, only some component equations within the equation system (4) are set equal to zero, in the sense of Eq. (15):

$$\sum_{i \in I} n_{ij} v_j(\mathbf{S}_{\text{slow}}, \mathbf{S}_{\text{fast}}) = 0\,, \tag{30}$$

where I is the set of fast variables, and $\mathbf{S}_{\text{slow}}$ and $\mathbf{S}_{\text{fast}}$ are the subvectors of the concentration vector that involve the slow and fast variables, respectively. This has to be interpreted correctly. It does not mean that the quasi-steady state variables are constant in time; they do change slowly, following the slow

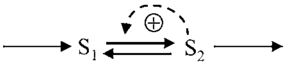


Figure 9 Simple reaction system with two internal metabolites and a positive feedback of S_2 on reaction 2. Reactions 1 and 3 are irreversible, while reaction 2 is reversible. The dashed arrow represents the activating regulatory interaction.

variables. They are linked with the slow variables by Eq. (30). Therefore the word "quasi" is used.

Equation (4) with part of it fulfilling Eq. (30) is called an algebro-differential equation system. There are various options for solving such equations systems numerically. If Eq. (30) can be transformed analytically so that $\mathbf{S}_{\text{fast}}$ can be written as a function of $\mathbf{S}_{\text{slow}}$, then this can be inserted into Eq. (4), so that this can be written as a differential equation system for $\mathbf{S}_{\text{slow}}$ and be integrated by any solver. Another option is to integrate the equations for the slow variables using the current values of the fast variables, then recompute the fast variables by solving Eq. (30). This needs to be either implemented individually or a hybrid algebraic-differential equation solver (such as DASSL by L. R. Petzold, e.g. Ref. [10]) can be used. A simulation package for cellular systems that is able to cope with algebro-differential equation systems is ProMoT/Diva [43].

The quasi-equilibrium approximation, in contrast, starts from the assumption that the fast enzyme reactions reach, after a short transient, a near-equilibrium state. That is, the mass-action ratio involving the concentrations of the metabolites participating in the fast reactions are approximately equal to the equilibrium constant:

$$\prod_{i=1}^{n} S_i^{n_{ij}} = q_j \,. \tag{31}$$

For numerical solution, similar methods as in the case of the quasi-steady-state approximation can be used.

Consider, for example, the simple reaction system in Figure 9, in which the second reaction is activated by its product. Such a positive feedback occurs in several biochemical reactions. For example, phosphofructokinase (see Figure 1) in yeast is activated by ADP [4], in many species also by AMP. (However, the mechanism underlying oscillations in glycolysis appears to be more complicated, see Section 4.4.) Let us assume that the reactions in the system in Figure 9 obey the rate laws:

$$v_1 = \text{const.} \tag{32a}$$

$$v_2 = (k_2 S_1 - k_2 S_2/q_2)\left(1 + \frac{S_2^2}{K_A}\right) \tag{32b}$$

$$v_3 = k_3 S_2 \,, \tag{32c}$$

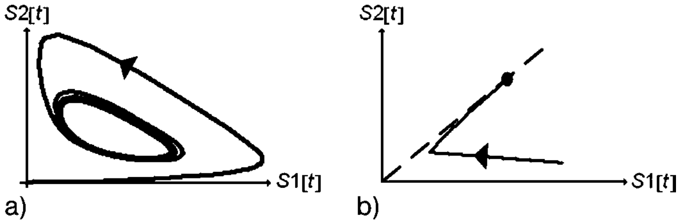


Figure 10 Dynamics of the reaction system shown in Figure 9 plotted in the phase plane of the two concentration variables. a) Case of oscillation; the trajectory tends to a stable limit cycle. b) Case of stable steady state (filled circle). Parameter values: $v_1 = 1$; $q_2 = 100$, $K_A = 0.1$; $k_3 = 1$ (A and B), $k_2 = 0.065$ (A), $k_2 = 100$ (B).

where k_2 and q_2 denote the rate constant and equilibrium constant, respectively, of reaction 2. K_A is called activation constant. The rate law for v_2 involves two factors. The first factor is an approximation of the reversible Michaelis–Menten rate law (7) (with $S = S_1$, $P = S_2$) for low concentrations. The second term describes the positive feedback. A term of the form (1 + concentration/parameter) is typical for enzyme kinetic rate laws describing activation (then this term is in the numerator) or inhibition (then it is in the denominator). Note that, when the concentration of the activator or inhibitor is zero, the original, unaffected rate law results. An oscillating system with a simpler version of the rate law for reaction 2, notably $v_2 = k_2 S_1 S_2^2$ has been proposed by Higgins [52] and Selkov [125] and is, therefore, nowadays called Higgins–Selkov oscillator. However, this rate law is less realistic then Eq. (32b) because it implies $v_2 = 0$ when $S_2 = 0$.

For a certain range of parameters, the system under study shows self-sustained oscillations (Figure 10A, see also Section 4.4). For other parameter values, the system runs to a stable steady state (Figure 10B). To show the usefulness of the quasi-equilibrium approximation, consider the case where the rate constant k_2 is large. The dynamics of relaxation to the steady state then shows two relatively distinct regions (Figure 10B). First, the curve (usually called the trajectory) representing the system dynamics in the space of variables goes from the initial conditions towards a straight line given by Eq. (31), which here reads:

$$\frac{S_2}{S_1} = q_2 \,. \tag{33}$$

In the second phase, the trajectory moves very close to this straight line. This can be explained as follows. When k_2 is large enough, i.e. when reaction 2 is fast enough, this reaction is, after a short transient, nearly at equilibrium even if it carries a nonzero flux. Accordingly, the concentration ratio approximately fulfils Eq. (33). When the quasi-equilibrium approximation is applied, it is often suitable to define combined ("pool") variables, such as $S_1 + S_2$ in our example [51]. Interestingly, they are the conservation quantities of the fast subsystem (see Section 3.1).

By the quasi-equilibrium and quasi-steady-state approximations, the dimension of the differential equation system is reduced. One might argue that with modern computers available, the size of equation systems is not an issue for numerical computation. However, these approximations also reduce the number of kinetic parameters. Since these parameters are often imperfectly known, this is a major advantage.

An intuitive way of implicitly using the quasi-steady-state or quasi-equilibrium approximations is the formulation of so-called skeleton models [50,126]. In writing down such models, one only considers the most essential variables, where the decision on what is essential is left to intuition or empirical knowledge. For example, Selkov [126] established a model of glycolysis with only two independent concentration variables (ATP and pool of trioses), while the skeleton model of [156] for the same system involves five variables.

4.2 Metabolic Databases

One of the major achievements of bioinformatics is the availability of large online databases. The fast access to genome-scale biological data of various types is of invaluable help in the modeling and simulation of processes in living cells, as stressed repeatedly in this book. As for the modeling of metabolic processes, especially enzyme and metabolic databases are used. Prominent examples are:

- ExPASy Enzyme (http://www.expasy.org/enzyme) [40]. This is a repository of various data on enzymes organized according to the EC (enzyme catalogue). Entries can be searched online, in a variety of ways, such as entering the EC number, official or alternative names, names of chemical compounds, or by browsing through the enzyme catalogue. The stored data comprise the reaction equation, protein sequence, alternative substrates, bibliographic links, etc., of the enzyme. Links to other databases and to biochemical pathway charts are included. The intention of the enzyme catalogue was to give a unique reference, and recommended name, to a distinct enzyme activity. Unfortunately, it does not use a controlled vocabulary for metabolite names, so that the name for the product of one reaction may not be the same as that used as the name for the substrate of the reaction that consumes it in the cell [55]. However, the data is curated, and there are mechanisms for submitting new entries and reporting errors.
- KEGG (http://www.genome.jp/kegg) [63]. Its full name being Kyoto Encyclopedia of Genes and Genomes, this database includes gene sequence information. It is, moreover, one of the most widely used metabolic databases

because it also includes a plethora of enzyme and metabolic pathway data. Interactive pathway charts allow the user to highlight all enzymes found so far in a specific species and click on specific enzymes to get information on these (such as reaction equations). KEGG also contains ample formation on diseases caused by enzyme deficiencies. Enzymes are distinguished by their EC number, where available, and metabolites participating in enzyme reactions. The reactions themselves are given supposedly unique identifiers and their own database entries. However, the metabolite identifiers have not avoided the problem of synonyms [55], and the incidence of other errors means that metabolic networks generated automatically from the databases are not fully connected and require manual correction. It is unclear whether there is an effective error-reporting procedure. Metabolic pathway information is also stored in the EMP database (http://www.empproject.com) [127].

- BRENDA (http://www.brenda.uni-koeln.de) [111]. This database can be compared to ExPASy Enzyme, but differs in one main aspect. It includes information on kinetic properties of enzymes, such as Michaelis–Menten constants, optima and intervals of pH and temperature, inhibitors, reaction mechanisms (e.g. ordered versus random mechanism), and subcellular localization. Therefore, it is particularly helpful as a basis for dynamic modeling.
- BioCyc (http://www.biocyc.org) [65]. This collection of databases has originated from an *E. coli* specific database, EcoCyc (which is still a part on its own within BioCyc) and is now devoted to many organisms. It involves both enzyme and pathway information, similar to KEGG, and its datamodel deals with the many-to-many relationships of genes, enzymes and reactions. Another part of that collection, MetaCyc, includes pathway information about more than 300 different organisms. EcoCyc is manually curated by experts in microbial metabolism; other organism specific data varies from being partially curated, to not significantly curated after automatic generation. There is a mechanism for reporting database errors.
- Reactome (http://www.reactome.org) [62] is a knowledgebase of core pathways and reactions in humans. In addition to curated entries about human biology, inferred orthologous events in 15 nonhuman species such as mouse, fly, yeast and *E. coli* are also available. In addition to metabolic pathways, signal transduction pathways such as those occurring in apoptosis, are included.

Several databases (e.g. KEGG and BioCyc) allow one to download data by ftp or other means. Thus, one might assume that an automated compilation of metabolic models on the basis of databases is feasible. This goal has not, however, been completely reached so far. First, the information is often error-prone, insecure and incomplete [8, 55]. To build a reliable model still requires checking the data in the original literature. At present, a genome-scale reconstruction of a metabolic network specifically assembled for one species is a nonautomated and iterative decision-making process requiring at least one researcher-year (e.g. Refs. [8, 37]). Second, ontological issues arise, as indicated above. In enzyme databases (as well as in the literature), substances are given at different levels of specificity. For example, in some databases, the term "branched-chain amino acids" is used in the context of aminotransferases, while in others the specific amino acids are mentioned separately. Another example is provided by alcohol dehydrogenase. The substrate of that enzyme is given as ethanol or primary alcohol. Thus, the additional information is needed that ethanol is a special case of primary alcohol. Nevertheless, online databases are extremely helpful for modeling because they provide an easy-to-get, comprehensive overview, which can be detailed and cross-checked later in the modeling process.

4.3 Example: Red Blood Cell Metabolism

The "guinea pig" of metabolic modeling is the red blood cell (*erythrocyte*) for various reasons. Mature erythrocytes in mammals are very simple cells. They do not contain any cell organelles such as mitochondria, not even a cell nucleus. The biological reason for this is that almost the entire cell volume is packed with hemoglobin, which serves for oxygen transport. The metabolism of these cells involves glycolysis (the most important pathway in these cells), the pentose phosphate pathway and part of purine metabolism. Figure 1 shows the main biochemical reactions involved in typical models of erythrocyte metabolism, notably glycolysis and the pentose phosphate pathway. Some models also include purine metabolism and the oxidation/reduction cycle of glutathione, which is driven by NADP/NADPH. Erythrocyte metabolism is even simpler than that of unicellular parasites such as *Mycoplasma*. Paradoxically, erythrocytes transport oxygen (as well as purine nucleotides), but are not "allowed" to use it for respiration. Another reason why erythrocytes are so intensively studied theoretically is that they have been thoroughly investigated experimentally due to easy accessibility. Moreover, they are of great medical importance.

The dominance of glycolysis in erythrocytes allows one to study a metabolic pathway of moderate size without much interference with other pathways. Depending on whether pyruvate or lactate are considered as end-products and whether the 2,3-bisphosphoglycerate bypass is included, glycolysis involves 10–13 reactions and about 15 concentration variables (Figure 1). In glycolysis, as in most other pathways, a timescale separation can be observed due to the presence of both slow and fast enzymes. This fact can be employed

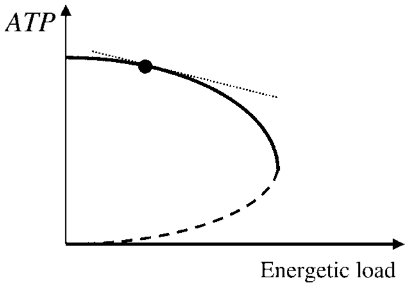


Figure 11 Schematic representation of the dependence of the stationary ATP concentration on the rate constant of ATP consumption (energetic load) in models of erythrocyte metabolism. Thick solid and dashed curves indicate stable and unstable steady states, respectively. The tangent (dotted line) to the branch of positive steady states characterizes the robustness of the system. Filled circle, *in vivo* point.

by applying the quasi-equilibrium approximation to about half of the reactions [113].

The most comprehensive kinetic models of erythrocyte metabolism so far have been developed in Refs. [61, 86]. These models include all the three parts of metabolism mentioned above. Subsequent models and simulations of erythrocyte metabolism [59,84] have focused on special aspects rather than extended the earlier models.

From a detailed kinetic model, many conclusions can be drawn. In the case of erythrocyte models, medically relevant conclusions concern, for example, the robustness of the ATP and NADH levels against changes in the energetic and oxidative loads. Energetic load means the consumption of ATP by diverse processes such as ion pumps, while oxidative load refers to the consumption of reduced redox equivalents such as NADH, NADPH, and glutathione by oxidizing reactions involving, for example, free radicals. Figure 11 shows a schematic picture of the dependence of the stationary ATP concentration on the rate constant of ATP consumption. Similar figures based on numerical simulations can be found, for example, in Ref. [51]. Note that for physiological values of the energetic load, there are two stable steady states – the trivial state with $ATP = 0$ and the *in vivo* state with positive ATP concentration. The reason for the existence of the trivial state is that glycolysis starts with a reaction consuming ATP (hexokinase, Figure 1). If no ATP is available, glycolysis is not "ignited" and cannot produce ATP.

Consider now the branch involving the *in vivo* state. If the energetic load becomes too high, the positive steady state disappears and the system "breaks down". The slope of the branch at the *in vivo* state can be interpreted as the robustness of the ATP producing system. Such figures (based on computational results) have also been presented for the dependence of redox equivalents on the energetic and oxidative loads [113, 114].

An important medical application of models of erythrocyte metabolism is the description and analysis of enzyme deficiencies. Many inherited diseases are based on complete or partial deficiencies of some enzymes, as mentioned in Section 2.1. An example relevant to red blood cells is provided by the sometimes occurring insufficient production of the enzyme hexokinase, which may lead to the disruption of these cells (hemolytic anemia). The impact of various enzyme deficiencies was assessed by simulations [112,114]. If (nearly) complete enzyme deficiencies are studied, network-based models can be used to make qualitative assertions [12,118].

4.4 Oscillations

Analysis of stationary states is the main paradigm in biochemical modeling. However, it is obvious that not all biological processes are stationary. The heartbeat, circadian clocks, cell division cycles, labor pain in childbirth and many other examples show oscillatory dynamics. There are, however, relatively few examples of oscillating metabolic processes (e.g. Ref. [45]). In most cases, the cause of oscillation is at a level other than the metabolic level, notably the genetic and hormonal levels, and electrophysiological effects (such as in the case of the heartbeat) can also play a role. A metabolic system relatively intensively studied in view of oscillations is the glycolytic pathway in yeast [5,77,104,125,126,156]. In these oscillations, the concentrations of all glycolytic intermediates move with the same frequency, yet partly in different phases.

However, glycolytic oscillations seem not to be very relevant under physiological conditions. Rather, they occur under special experimental conditions, e.g. after addition of cyanide during the diauxic shift from glucose consumption to ethanol consumption. In general, biochemical oscillations appear to be the exception rather than the rule. Most biochemical systems subsist in stable steady states, due to the stabilizing effect of usual chemical kinetics and probably because metabolic oscillations are rarely physiologically favorable. Nevertheless, glycolytic oscillations are of high academic interest because they can be used as an appropriate example to which the toolbox of theoretical methods can be applied. For example, the parameter values at which the Higgins–Selkov system (see Section 4.1) starts oscillating can be calculated analytically (e.g. Refs. [51,70]).

A physiologically very important oscillating system where biochemical processes are involved is the so-called mitotic oscillator, that is the system underlying the cell division cycle (for a model, see Ref. [87]). Another intensely studied case is cAMP oscillations in *Dictyostelium discoideum* (for a model, see Ref. [48]). However, that system rather belongs to signal transduction because cAMP oscillations serve for communication among these amoebae before formation of the fruiting body. A phenomenon studied even more intensely than glycolytic oscillations is that of intracellular calcium oscillations [28, 45, 122, 134]. These play an important role in signaling as well because calcium is a second messenger (see Chapter 22).

Different types of oscillations need to be distinguished depending on their stability. The biologically most relevant type is that of asymptotically stable oscillations; after a small fluctuation, the system returns to the same time course (though perhaps with another phase of oscillation). This type is usually called stable limit cycle (see Figure 10B). A second type is that of marginally stable oscillations. Here, a small fluctuation leads to a permanent shift in the maximum and minimum values of the oscillatory variables. A prominent example is the predator–prey system in population dynamics first analyzed by Lotka and Volterra (e.g. Ref. [85]). In linear differential equation systems, limit cycles cannot occur while marginally stable oscillations can. This is because the solutions of such equations are exponential functions. The real part of the exponents is either positive (which leads to a permanent increase in amplitude until infinity), zero (which leads to a marginally stable amplitude) or negative (which leads to damped oscillations). Thus, limit cycles can occur only in nonlinear systems. In fact, the nonlinearity must usually be rather strong. The essential nonlinearity in Eq. (32b) is $S_1S_2^2$. It can be shown by analyzing the Jacobian matrix that the term S_1S_2 would not suffice to produce oscillations. In the model of calcium oscillations developed by Somogyi and Stucki [134], for example, it is a Hill kinetics of the fourth degree. One oscillatory mechanism is positive feedback (e.g. due to autocatalysis or product activation). Most models of calcium oscillations are based on the positive feedback of cytosolic calcium on its release from the endoplasmic reticulum. For a long time, positive feedback by product activation of the glycolytic enzyme phosphofructokinase (see Figure 1) has also been discussed as an oscillatory mechanism (see Section 4.1). Apart from this, a positive feedback due to stoichiometric effects, notably the consumption of ATP at the upper end of glycolysis and production of ATP at the lower end, was also considered [125, 126]. However, the mechanism underlying glycolytic oscillations appears to be more complicated than either of these hypotheses [77].

In continuous modeling, autonomous oscillations cannot occur in a 1-D system. If for a 1-D equation, $\mathrm{d}x/\mathrm{d}t = f(x)$, the curve $x(t)$ were to have a monotonic increasing part and a monotonic decreasing part, it would need to pass a point where the time derivative $\mathrm{d}x/\mathrm{d}t$ equals zero. At this point, however, $f(x)$ is zero, so that x remains constant and cannot, hence, decrease – a contradiction (see also Ref. [137]). Accordingly, all continuous models of the above-mentioned types of biological oscillations are at least 2-D.

4.5 Whole-cell Modeling

A current trend in metabolic modeling is to aim at simulating the metabolism of entire cells (e.g. Ref. [142]). This is motivated by the breathtaking achievements in sequencing whole genomes. Thus, many groups are making attempts to perform modeling on a genome scale. This is done either by scaling up models on the basis of existing software tools or by creating new software tools specially designed for large systems. Projects in the latter direction are Electronic cell [139], Virtual Cell [132] and Silicon Cell [133]. Virtual Cell has the special feature that it allows spatial modeling. It provides a formal framework for simulating biochemical, electrophysiological and transport processes, while considering the subcellular localization of the various substances. However, most systems simulated by packages for "whole-cell modeling" so far have a size that can be treated with other simulation packages (such as GEPASI) as well.

There has been much controversy in the literature about the pros and cons of whole-cell modeling. While this issue is largely beyond the scope of a textbook, we wish to discuss at least some points. Any model is a simplified representation of some aspect of reality and usually serves a practical purpose. For example, a model can be established to answer the question what mechanism allows for calcium oscillations in the cell. A 2-D differential equation system is sufficient to provide the answer to this specific question: the positive feedback exerted by calcium-induced calcium release provides such a mechanism (e.g. Refs. [28, 45, 122]). The idea of whole-cell modeling is based on a different "philosophy" – rather than dealing with a specific question and restricting the analysis to some part or aspect from the outset, a comprehensive picture of the cell in its entirety is aimed at, from which specific answers are to be derived later. Such a model would integrate the available knowledge of the structure of the metabolic network and the parameters of the system (such as enzyme kinetic parameters), and could establish whether it is sufficient to account for the observed metabolic characteristics and responses of the cell. Whether this would amount to an explanation is open to question, since it would still be necessary to analyze the model further to determine which structural features and kinetic parameters are particularly influential for specific behaviors. The predictive power of such a detailed and comprehensive model could potentially be greater, since the consequences of any intervention at the molecular level (such as mutation or enzyme inhibition by a drug) could be followed through the whole system.

It has been questioned whether such a comprehensive, perfect picture can be established. Major problems arise from three sources:

(i) For the vast majority of enzymes, the kinetic parameters such as maximal velocities (which are proportional to *in vivo* enzyme concentrations) are unknown. Therefore, approximation methods such as the quasi-steady-state approximation are often used, and most models are restricted to small parts of metabolism. In fact, each enzyme-kinetic rate law is based on an approximation (see Section 2.4).

(ii) Computational problems. While modern computer technology allows one to solve thousands of differential equations simultaneously and deal with large linear programming problems, there are some questions in metabolic modeling that are computationally hard. This concerns, in particular, problems related to combinatorial explosion, such as in the computation of extreme pathways or elementary modes.

(iii) Sometimes, the objection against whole-cell models is raised that even if all data were known and the respective computer simulations were feasible, the huge amount of output data are hard to understand and interpret. This is to do with constraints in the human mind. However, analyzing the output data of computations can often be facilitated by automated extraction methods. Computer routines can help extract the features of interest (e.g. all elementary modes producing a given substance).

The problems in establishing dynamic models of the metabolism of whole cells can be illustrated by the huge amount of work that has been spent over about 40 years on the modeling and simulation of a very simple cell – the human erythrocyte [50,51,59,61,84,86,113]. In spite of these enormous efforts, we are far from being able to simulate all aspects of erythrocyte metabolism. Thus, simulating the entire metabolism of more complex cells is unlikely to be attained in the near future.

On the other hand, laboratory automation of the type that accelerated genome sequencing and that is applied in the pharmaceutical industry for high-throughput screening could be used to acquire enzyme kinetic parameters much more rapidly than in the past. Of course, it is of great academic and practical interest to scale up metabolic models. For example, testing the effect of a drug *in silico* at the whole-cell level can be extremely useful. The academic quest for whole-cell or even whole-organism models can be compared to the striving of mankind to achieve nuclear fusion or flying to Mars. Indeed, the development of computable models of organisms has been nominated as one of the current grand challenges in computing (http://www.ukcrc.org.uk/grand_challenges/index.cfm).

An approach that is, in a sense, in between small (or even minimal) models and whole-cell models is to simulate only special aspects of whole cells, such as their network properties, rather than all their dynamic properties. However, even this aim is difficult to achieve because of combinatorial complexity. For example, computing the elementary modes for the whole of metabolism is impossible at present due to combinatorial explosion, even for microorganisms as simple as *Mycoplasma pneumoniae*. Currently available software tools for network analysis are able to cope with systems of about 200 reactions. Stelling and coworkers [135] computed the elementary modes for a model containing 110 reactions, covering a considerable part of central metabolism of *E. coli*. When four different substrates are allowed simultaneously, this gives rise to about half a million modes. The current version, METATOOL 5.0 (http://pinguin.biologie.uni-jena.de/bioinformatik/networks/index.html) is able to cope with a system extended to 112 reactions, computing 2 450 787 modes [146]. The number of elementary modes depends very much on the special properties of the network. In particular, adding reactions connecting to external metabolites (exchange reactions) usually increases the number of modes much more than adding internal reactions.

The above-mentioned *E. coli* system [135] was used to predict the viability of single mutants, with 90% of the predictions being consistent with experimental data. An even larger system of *E. coli* metabolism with 720 reactions was analyzed in Ref. [25]. The viability of single mutants was predicted by FBA rather than by elementary-modes analysis, with 86% of the predictions being correct. In fact, FBA is sufficient to predict viability because, if the flux distribution with the best biomass-over-substrate yield is not able to sustain growth, then no flux distribution is. FBA (which is based on linear programming) can cope with larger systems than elementary-modes analysis because not all possible extreme flux situations need be enumerated. On the other hand, it gives a less comprehensive picture of the system's capabilities. The extreme-pathway analyses in Refs. [91, 103] (see Section 3.6) have been called "genome-wide" by their authors. However, combinatorial explosion was avoided by considering one product (amino acid in that case) at a time and macromolecule synthesis was excluded.

One of the largest metabolic networks so far reconstructed from genome data was presented in [37]. It comprises 1175 enzymatic reactions and 584 metabolites in *S. cerevisiae*. Carrying out a linear programming analysis (FBA) is feasible for that network. Applying FBA to the reconstructed network, Förster and coworkers [37] predicted the metabolic capabilities of *S. cerevisiae* when glucose is the sole carbon source and compared them with *E. coli*. It turned out that *S. cerevisiae* can synthesize the 12 most important precursor metabolites and the 20 proteinogenic amino acids more efficiently than *E. coli.*

The *S. cerevisiae* network reconstructed in Ref. [37], the *E. coli* network compiled in Ref. [25] and a network of *Helicobacter pylori* metabolism comprising 389 reactions [109] were used for a flux-coupling analysis [11]. Flux coupling analysis is part of network analysis and means that the coupling between directionalities or values of fluxes in steady states is computed. For example, in an unbranched reaction chain, in which at least one reaction is irreversible, all fluxes must be equal and their directionality is determined by the irreversible reaction. Flux-coupling analysis allows, in particular, the detection of infeasible fluxes (called blocked reactions in Ref. [11]). These are fluxes that are always zero at steady state. It was determined that 10, 14 and 29% of the reactions in the three above-mentioned networks, respectively, are blocked reactions. This may, however, be due to incompleteness of the reconstructed networks.

5 Conclusions

In this chapter, an outline of the modeling and simulation of metabolic systems has been given. Of course, the field of metabolic modeling is much broader than could be sketched here. We should at least mention metabolic control analysis [5, 19, 32, 50, 51], in which the control by particular enzymes over systemic properties such as steady-state fluxes is analyzed.

In addition to models of erythrocyte metabolism, many models of parts of metabolism in other cell types have been established. For example, a model and computer simulations of the threonine-synthesis pathway in *E. coli* were presented by Chassagnole and coworkers [14, 15]. As pointed out above, glycolysis in yeast has been studied intensely in view of oscillations. Nevertheless, simulations of this pathway in the physiologically more relevant regime of steady state have been presented as well [39, 140]. A number of them can be examined, and run, at a web repository curated by J. L. Snoep (http://jjj.biochem.sun.ac.za).

An important current trend is network modeling (constraint-based modeling), as outlined in Section 3. As shown by the CAM plant and TCA cycle examples in Section 3.5, relevant conclusions can be drawn by this analysis without the need for kinetic data. The algorithms and mathematical theory in this field are permanently refined [38,66,144], and applied to biotechnological [13,101,115,119,120,123,136] and medical problems [12,118,151] as well as to functional genomics [20, 30]. A large body of literature on FBA has emerged (some papers are cited in Section 3.7). Moreover, there are a number of approaches to explaining the properties of metabolic networks by optimality principles other than maximum yield, e.g. maximum pathway flux [2, 51, 147] and also flux minimization [54]. A modern line of research is based on game-theoretical approaches to studying the evolution of metabolic pathways [3, 47, 96, 99].

Mathematical modeling and simulation in all fields of science have several objectives. One goal is the explanation and better understanding of experimental observations. This goal has been reached many times in metabolic modeling. For example, modeling has allowed us to find the mechanistic bases of glycolytic and calcium oscillations (see Section 4.4). Another example is provided by the elementary modes of the CAM plant metabolism, which allowed a systematic classification of experimental data (see Section 3.5). A more ambitious goal is to predict some novel phenomenon, which has not been observed before. Theoretical physics has often achieved this (e.g. the prediction of the positron). Theoretical biology is less predictive so far. Nevertheless, some success stories exist. For example, the theoretically predicted optimal time course of gene expression in an unbranched metabolic pathway [69] could be confirmed experimentally [158]. The predictions of FBA for *E. coli* could be verified in adaptive evolution experiments [36, 57]. Another example is the catabolic pathway in hungry *E. coli* mentioned in Section 3.3. In conclusion, it can be said that theoretical biology and bioinformatics will shift more and more towards predictive sciences.

Acknowledgments

D. F. thanks Alyson Tobin and Anne Borland for their suggestions for starch and hexose-producing CAM plants. His research on elementary-modes analysis has been supported by the BBSRC and Oxford Brookes University. Moreover, the authors wish to thank Beate Knoke for technical support with the figures, and Chris Lauber, Herbert Sauro and Ernst Günter Schukat-Talamazzini for helpful discussions.

References

1 Alter, O., P. O. Brown and D. Botstein. 2003. Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. Proc. Natl Acad. Sci. USA **100**: 3351–6.

2 Angulo-Brown, F., M. Santillan and E. Calleja-Quevedo. 1995. Thermodynamic optimality in some biochemical reactions. Nuovo Cim. **17**: D87–90.

3 Anten, N. P. 2005. Optimal photosynthetic characteristics of individual plants in vegetation stands and implications for species coexistence. Ann. Bot. (Lond.) **95**: 495–506.

4 Bär, J., G. Hübner and G. Kopperschläger. 1986. Study on the initial kinetics of yeast phosphofructokinase by stopped-flow measurements. Eur. J. Biochem. **156**: 311–5.

5 Bier, M., B. Teusink, B. N. Kholodenko and H. V. Westerhoff. 1996. Control analysis of glycolytic oscillations. Biophys. Chem. **62**: 15–24.

6 Borland, A. M. and A. N. Dodd. 2002. Carbohydrate partitioning in Crassulacean acid metabolism plants: reconciling potential conflicts of interest. Funct. Plant Biol. **29**: 707–16.

7 Borland, A. M., H. Griffiths, M. S. J. Broadmeadow, M. C. Fordham and C. Maxwell. 1994. Carbon isotope composition of biochemical fractions and the regulation of carbon balance in leaves of the C3-Crassulacean acid metabolism intermediate *Clusia minor L.* growing in Trinidad. Plant Physiol. **106**: 493–501.

8 Borodina, I., P. Krabben and J. Nielsen. Genome-scale analysis of *Streptomyces coelicolor* A3(2) metabolism. Genome Res. **15**: 820–9.

9 Braess, D. 1968. Über ein Paradoxon aus der Verkehrsplanung. Unternehmensforschung **12**: 258–68.

10 Breman, K. E., S. L. Campbell and L. R. Petzold. 1989. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Elsevier, New York, NY.

11 Burgard, A. P., E. V. Nikolaev, C. H. Schilling and C. D. Maranas. 2004. Flux coupling analysis of genome-scale metabolic network reconstructions. Genome Res. **14**: 301–12.

12 Çak*ı*r, T., C. S. Tacer and K. Ö. Ülgen. 2004. Metabolic pathway analysis of enzyme-deficient human red blood cells. BioSystems **78**: 49–67.

13 Carlson, R., D. Fell and F. Srienc. 2002. Metabolic pathway analysis of a recombinant yeast for rational strain development. Biotechnol. Bioeng. **79**: 121–34.

14 Chassagnole, C., B. Raïs, E. Quentin, D. A. Fell and J.-P. Mazat. 2001. An integrated study of threonine-pathway enzyme kinetics in *Escherichia coli.* Biochem. J. **356**: 415–23.

15 Chassagnole, C., D. A. Fell, B. Rais, B. Kudla and J.-P. Mazat. 2001. Control of the threonine–synthesis pathway in *Escherichia coli*: a theoretical and experimental approach. Biochem. J. **356**: 433–44.

16 Christopher, J. T. and J. A. M. Holtum. 1996. Patterns of carbon partitioning in leaves of Crassulacean acid metabolism species during deacidification. Plant Physiol. **112**: 393–9.

17 Clarke, B. L. 1981. Complete set of steady states for the general stoichiometric dynamical system. J. Chem. Phys. **75**: 4970–9.

18 Cohen, S. D. and A. C. Hindmarsh. 1996. CVODE, a stiff/nonstiff ODE solver in C. Comput. Phys. **10**: 138–43.

19 Cornish-Bowden, A. 1995. *Fundamentals of Enzyme Kinetics*. Portland Press, London.

20 Dandekar, T., S. Schuster, B. Snel, M. Huynen and P. Bork. 1999. Pathway alignment: application to the comparative analysis of glycolytic enzymes. Biochem. J. **343**: 115–24.

21 de Hollander, J. A. 1994. Potential metabolic limitations in lysine production by *Corynebacterium glutamicum* as revealed by metabolic network analysis. Appl. Microbiol. Biotechnol. **42**: 508–15.

22 Dodd, A. N., A. M. Borland, R. P. Haslam, H. Griffiths and K. Maxwell. 2002. Crassulacean acid metabolism: plastic, fantastic. J. Exp. Bot. **53**: 569–80.

23 Edwards, G. E., J. G. Foster and K. Winter. 1982. Activity and compartmentation of enzymes of carbon metabolism in CAM plants. In Ting, I.P. and M. Gibbs (eds.), *Crassulacean Acid Metabolism*. American Society of Plant Physiologists, Rockville, MD: 92–111.

24 Edwards, J. S. and B. O. Palsson. 1999. Systems properties of the *Haemophilus influenzae* Rd metabolic genotype. J. Biol. Chem. **274**: 17410–6.

25 Edwards, J.S. and B. O. Palsson. 2000. The *Escherichia coli* MG1655 *in silico* metabolic genotype: Its definition, characteristics, and capabilities. Proc. Natl Acad. Sci. USA **97**: 5528–33.

26 Edwards, J. S., R. U. Ibarra and B. O. Palsson. 2001. *In silico* predictions of *Escherichia coli* metabolic capabilities are consistent with experimental data. Nat. Biotechnol. **19**: 125–30.

27 Edwards, J. S., R. Ramakrishna, and B. O. Palsson. 2002. Characterizing the metabolic phenotype: a phenotype phase plane analysis. Biotechnol. Bioeng. **77**: 27–36.

28 Falcke, M. 2004. Reading the patterns in living cells – the physics of Ca^{2+} signaling. Adv. Phys. **53**: 255–440.

29 Famili, I. and B. O. Palsson. 2003. The convex basis of the left null space of the stoichiometric matrix leads to the definition of metabolically meaningful pools. Biophys. J. **85**: 16–26.

30 Famili, I., J. Förster, J. Nielsen and B. O. Palsson. 2003. *Saccharomyces cerevisiae* phenotypes can be predicted by using constraint-based analysis of a

genome-scale reconstructed metabolic network. Proc. Natl Acad. Sci. USA **100**: 13134–9.

31 FELL, D. A. 1990. Substrate cycles: theoretical aspects of their role in metabolism. Commun. Theor. Biol. **6**: 1–14.

32 FELL, D. 1997. *Understanding the Control of Metabolism.* Portland Press, London.

33 FELL, D. A. AND J. R. SMALL. 1986. Fat synthesis in adipose tissue. An examination of stoichiometric constraints. Biochem. J. **238**: 781–86.

34 FISCHER, E. AND U. SAUER. 2003. A novel metabolic cycle catalyzes glucose oxidation and anaplerosis in hungry *Escherichia coli.* J. Biol. Chem. **278**: 46446–51.

35 FISCHER, E. AND U. SAUER. 2005. Large-scale *in vivo* flux analysis shows rigidity and suboptimal performance of *Bacillus subtilis* metabolism. Nat. Genet. **37**: 636–40.

36 FONG, S. S. AND B. O. PALSSON. 2004. Metabolic gene-deletion strains of *Escherichia coli* evolve to computationally predicted growth phenotypes. Nat. Genet. **36**: 1056–8.

37 FÖRSTER, J., I. FAMILI, P. FU, B. Ø. PALSSON AND J. NIELSEN. 2003. Genome-scale reconstruction of the *Saccharomyces cerevisiae* metabolic network. Genome Res. **13**: 244–53.

38 GAGNEUR, J. AND S. KLAMT. 2004. Computation of elementary modes: a unifying framework and the new binary approach. BMC Bioinformatics **5**: 175.

39 GALAZZO, J. L. AND J. E. BAILEY. 1990. Fermentation pathway kinetics and metabolic flux control in suspended and immobilized *Saccharomyces cerevisiae*. Enzyme Microb. Technol. **12**: 162–72.

40 GASTEIGER, E., A. GATTIKER, C. HOOGLAND, I. IVANYI, R. D. APPEL AND A. BAIROCH. 2003. ExPASy: the proteomics server for in-depth protein knowledge and analysis. Nucleic Acids Res. **31**: 3784–8.

41 GEAR, C. W. 1971. *Numerical Initial Value Problems in Ordinary Differential Equations*. Prentice-Hall, Englewood Cliffs, NJ.

42 GIAEVER, G., A. M. CHU, L. NI, et al. 2002. Functional profiling of the *Saccharomyces cerevisiae* genome. Nature **418**: 387–91.

43 GINKEL, M., A. KREMLING, T. NUTSCH, R. REHNER AND E. D. GILLES. 2003. Modular modeling of cellular systems with ProMoT/Diva. Bioinformatics **19**: 1169–76.

44 GOFFRINI, P., I. FERRERO AND C. DONNINI. 2002. Respiration-dependent utilization of sugars in yeasts: a determinant role for sugar transporters. J. Bacteriol **184**: 427–32.

45 GOLDBETER, A. 1996. *Biochemical Oscillations and Cellular Rhythms*. Cambridge University Press, Cambridge.

46 GORYANIN, I., T. C. HODGMAN AND E. SELKOV. 1999. Mathematical simulation and analysis of cellular metabolism and regulation. Bioinformatics **15**: 749–58.

47 GREIG, D. AND M. TRAVISANO. 2004. The Prisoner's Dilemma and polymorphism in yeast *SUC* genes. Proc. R. Soc. Lond. B **271 (Suppl. 3)**: S25–6.

48 HALLOY, J., J. LAUZERAL AND A. GOLDBETER. 1998. Modeling oscillations and waves of cAMP in *Dictyostelium discoideum* cells. Biophys. Chem. **72**: 9–19.

49 HAUSLER, R. E., B. BAUR, J. SCHARTE, et al. 2000. Plastidic metabolite transporters and their physiological functions in the inducible Crassulacean acid metabolism plant *Mesembryanthemum crystallinum*. Plant J. **24**: 285–96.

50 HEINRICH, R., S. M. RAPOPORT AND T. A. RAPOPORT. 1977. Metabolic regulation and mathematical models. Prog. Biophys. Mol. Biol. **32**: 1–82.

51 HEINRICH, R. AND S. SCHUSTER. 1996. *The Regulation of Cellular Systems*. Chapman & Hall, New York, NY.

52 HIGGINS, J. 1964. A chemical mechanism for oscillation of glycolytic intermediates in yeast cells. Proc. Natl Acad. Sci. USA **51**: 989–94.

53 HOFMEYR, J. H. S. AND A. CORNISH-BOWDEN. 1997. The reversible Hill equation: how to incorporate cooperative enzymes into metabolic models. Comput. Appl. Biosci. **13**: 377–85.

54 HOLZHÜTTER, H. G. 2004. The principle of flux minimization and its application to estimate stationary fluxes in metabolic networks. Eur. J. Biochem. **271**: 2905–22.

55 HORNE, A. B., T. C. HODGMAN, H. D. SPENCE AND A. R. DALBY. 2004. Constructing an enzyme-centric view of metabolism. Bioinformatics **20**: 2050–55.

56 HUCKA, M., A. FINNEY, H. M. SAURO, et al. 2003. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics **19**: 524–31.

57 IBARRA, R. U, J. S. EDWARDS AND B. O. PALSSON. 2002. *Escherichia coli* K-12 undergoes adaptive evolution to achieve in silico predicted optimal growth. Nature **420**: 186–9.

58 INDERLIED, C. B. AND P. S. SYPHERD. 1978. Glucose metabolism and dimorphism in Mucor. J. Bacteriol **133**: 1282–6.

59 JAMSHIDI, N., J. S. EDWARDS, T. FAHLAND, G. M. CHURCH AND B. O. PALSSON. 2001. Dynamic simulation of the human red blood cell metabolic network. Bioinformatics **17**: 286–7.

60 JEONG, H., B. TOMBOR, R. ALBERT, Z. N. OLTVAI AND A. L. BARABASI. 2000. The large-scale organization of metabolic networks. Nature **407**: 651–4.

61 JOSHI, A. AND B. O. PALSSON. 1989. Metabolic dynamics in the human red cell. Part I. A comprehensive kinetic model. J. Theor. Biol. **141**: 515–28.

62 JOSHI-TOPE, G., M. GILLESPIE, I. VASTRIK, P. D'EUSTACHIO, E. SCHMIDT AND B. DE BONO. 2005. Reactome: a knowledgebase of biological pathways. Nucleic Acids Res. **33**: D428–32.

63 KANEHISA M., S. GOTO, S. KAWASHIMA, Y. OKUNO AND M. HATTORI. 2004. The KEGG resource for deciphering the genome. Nucleic Acids Res. **32**: D277–80.

64 KAPPLER, O., P. H. JANSSEN, J. U. KREFT AND B. SCHINK. 1997. Effects of alternative methyl group acceptors on the growth energetics of the *O*-demethylating anaerobe *Holophaga foetida.* Microbiology **143**: 1105–14.

65 KARP, P. D., C. A. OUZOUNIS, C. MOORE-KOCHLACS, L. GOLDOVSKY, P. KAIPA AND D. AHREN. 2005. Expansion of the BioCyc collection of pathway/genome databases to 160 genomes. Nucleic Acids Res. **33**: 6083–9.

66 KLAMT, S. AND E. D. GILLES. 2004. Minimal cut sets in biochemical reaction networks. Bioinformatics **20**: 226–34.

67 KLAMT, S. AND J. STELLING. 2003. Two approaches for metabolic pathway analysis? Trends Biotechnol. **21**: 64–9.

68 KLAMT, S., J. STELLING, M. GINKEL, E. D.GILLES. 2003. FluxAnalyzer: exploring structure, pathways, and fluxes in balanced metabolic networks by interactive flux maps. Bioinformatics **19**: 261–9.

69 KLIPP, E., R. HEINRICH AND H. G. HOLZHÜTTER. 2002. Prediction of temporal gene expression. Metabolic optimization by re-distribution of enzyme activities. Eur. J. Biochem. **269**: 5406–13.

70 KLIPP, E., R. HERWIG, A. KOWALD, C. WIERLING AND H. LEHRACH. 2005. *Systems Biology in Practice*. Wiley-VCH, Weinheim.

71 LAND, G. A., W. C. MCDONALD, R. L. STJERNHOLM AND L. FRIEDMAN. 1975. Factors affecting filamentation in *Candida albicans*: changes in respiratory activity of *Candida albicans* during filamentation. Infect. Immun. **12**: 119–27.

72 LAY, D. C. 2002. *Linear Algebra and its Applications*. Addison-Wesley, Boston, MA.

73 LEE, D. D. AND H. S. SEUNG. 1999. Learning the parts of objects by non-negative matrix factorization. Nature **401**: 788–91.

74 LIAO, J. C., S.-Y. HOU AND Y.-P. CHAO. 1996. Pathway analysis, engineering, and physiological considerations for redirecting central metabolism. Biotechnol. Bioeng. **52**: 129–40.

75 LOVERING, R. M., N. C. PORTER AND R. J. BLOCH. 2005. The muscular dystrophies: from genes to therapies. Phys. Ther **85**: 1372–88.

76 MA, H. W. AND A. P. ZENG. 2003. Reconstruction of metabolic networks from genome data and analysis of their

global structure for various organisms. Bioinformatics **19**: 270–7.

77 MADSEN, M. F, S. DANO AND P. G. SORENSEN. 2005. On the mechanisms of glycolytic oscillations in yeast. FEBS J. **272**: 2648–60.

78 MAHADEVAN, R. AND C. H. SCHILLING. 2003. The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. Metab. Eng. **5**: 264–76.

79 MAVROVOUNIOTIS, M. L., G. STEPHANOPOULOS AND G. STEPHANOPOULOS. 1990. Computer-aided synthesis of biochemical pathways. Biotechnol. Bioeng. **36**: 1119–32.

80 MENDES, P. 1997. Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. Trends Biochem. Sci. **22**: 361–3.

81 MENDES P. AND D. B. KELL. 2001. MEG (Model Extender for Gepasi): a program for the modelling of complex, heterogeneous, cellular systems. Bioinformatics **17**: 288–9.

82 MICHAELIS, L. AND M. L. MENTEN. 1913. Die Kinetik der Invertinwirkung. Biochem. Z. **49**: 333–69.

83 MOROHASHI, M., A. E. WINN, M. T. BORISUK, H. BOLOURI, J. DOYLE AND H. KITANO. 2002. Robustness as a measure of plausibility in models of biochemical networks. J. Theor. Biol. **216**: 19–30.

84 MULQUINEY, P. J. AND P. W. KUCHEL. 1999. Model of 2,3-bisphosphoglycerate metabolism in the human erythrocyte based on detailed enzyme kinetic equations: computer simulation and metabolic control analysis. Biochem. J. **342**: 597–604.

85 MURRAY, J. D. 2002. *Mathematical Biology*, Vol. I. Springer, New York, NY.

86 NI, T. C. AND M. A. SAVAGEAU. 1996. Application of biochemical systems theory to metabolism in human red blood cells. Signal propagation and accuracy of representation. J. Biol. Chem. **271**: 7927–41.

87 NOVAK, B., Z. PATAKI, A. CILIBERTO AND J. J. TYSON. 2001. Mathematical model of the cell division cycle of fission yeast. Chaos **11**: 277–86.

88 NOŽIČKA, F., J. GUDDAT, H. HOLLATZ AND B. BANK. 1974. *Theorie der linearen parametrischen Optimierung*. Akademie-Verlag, Berlin.

89 NUÑO, J. C., I. SÁNCHEZ-VALDENEBRO, C. PÉREZ-IRATXETA, E. MELÉNDEZ-HEVIA AND F. MONTERO. 1997. Network organization of cell metabolism: monosaccharide interconversion. Biochem. J. **324**: 103–11.

90 OLIVIER, B. G., J. M. ROHWER AND J. H. HOFMEYR. 2005. Modelling cellular systems with PySCeS. Bioinformatics **21**: 560–1.

91 PAPIN, J. A., N. D. PRICE, J. S. EDWARDS AND B. O. PALSSON. 2002. The genome-scale metabolic extreme pathway structure in *Haemophilus influenzae* shows significant network redundancy. J. Theor. Biol. **215**: 67–82.

92 PAPIN, J. A., J. STELLING, N. D. PRICE, S. KLAMT, S. SCHUSTER AND B. O. PALSSON. 2004. Comparison of network-based pathway analysis methods. Trends Biotechnol. **22**: 400–5.

93 PAPOUTSAKIS, E. T. 1984. Equations and calculations for fermentations of butyric-acid bacteria. Biotechnol. Bioeng. **26**: 174–87.

94 PETTINEN, A., T. AHO, O.-P. SMOLANDER, T. MANNINEN, A. SAARINEN, K.-L. TAATTOLA, O. YLI-HARJA AND M.-L. 2005. Linne. Simulation tools for biochemical networks: evaluation of performance and usability. Bioinformatics **21**: 357–63.

95 PETZOLD, L. 1983. Automatic selection of methods for solving stiff and nonstiff systems of ordinary systems of differential equations. SIAM J. Sci. Stat. Comput. **4**: 136–48.

96 PFEIFFER, T. AND S. BONHOEFFER. Evolution of crossfeeding in microbial populations. Am. Nat. **163**: E126–35

97 PFEIFFER, T., I. SÁNCHEZ-VALDENEBRO, J. C. NUÑO, F. MONTERO AND S. SCHUSTER. 1999. METATOOL: for studying metabolic networks. Bioinformatics **15**: 251–7.

98 PFEIFFER, T., S. SCHUSTER AND S. BONHOEFFER. 2001. Cooperation and competition in the evolution of ATP

producing pathways. Science **292**: 504–7.

99 PFEIFFER, T. AND S. SCHUSTER: 2005. Game-theoretical approaches to studying the evolution of biochemical systems. Trends Biochem. Sci. **30**: 20–5.

100 POOLMAN, M. G., D. A. FELL AND C. A. RAINES. 2003. Elementary modes analysis of photosynthate metabolism in the chloroplast stroma. Eur. J. Biochem. **270**: 430–9.

101 POOLMAN, M. G., H. ASSMUSS AND D. A. FELL. 2004. Applications of metabolic modelling to plant metabolism. J. Exp. Bot **55**: 1177–86.

102 POOLMAN, M. G., K. V. VENKATESH, M. K. PIDCOCK AND D. A. FELL. 2004. A method for the determination of flux in elementary modes, and its application to *Lactobacillus rhamnosus*. Biotechnol. Bioeng. **88**: 601–12.

103 PRICE, N. D., J. A. PAPIN AND B. O. PALSSON. 2002. Determination of redundancy and systems properties of the metabolic network of *Helicobacter pylori* using genome-scale extreme pathway analysis. Genome Res. **12**: 760–9.

104 RICHARD, P., J. A. DIDERICH, B. M. BAKKER, B. TEUSINK, K. VAN DAM AND H. V. WESTERHOFF. 1994. Yeast cells with a specific cellular make-up and an environment that removes acetaldehyde are prone to sustained glycolytic oscillations. FEBS Lett. **341**: 223–6.

105 SAURO, H. M. 1993. SCAMP: a general-purpose simulator and metabolic control analysis program. Comput. Appl. Biosci. **9**: 441–50.

106 SAURO, H. M., M. HUCKA, A. FINNEY, C. WELLOCK, H. BOLOURI, J. DOYLE AND H. KITANO. 2003. Next generation simulation tools: the Systems Biology Workbench and BioSPICE integration. OMICS **7**: 355–72.

107 SAVAGEAU, M. A. 1976. *Biochemical Systems Analysis*. Addison-Wesley, Reading, MA.

108 SCHILLING, C. H., D. LETSCHER AND B. O. PALSSON. 2000. Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. J. Theor. Biol. **203**: 229–48.

109 SCHILLING, C. H., M. W. COVERT, I. FAMILI, G. M. CHURCH, J. S. EDWARDS AND B. O. PALSSON. 2002. Genome-scale metabolic model of *Helicobacter pylori* 26695. J. Bacteriol. **184**: 4582–93.

110 SCHNEIDER, K. R. AND T. WILHELM. 2000. Model reduction by extended quasi-steady-state approximation. J. Math Biol. **40**: 443–50.

111 SCHOMBURG I, A. CHANG, C. EBELING, M. GREMSE, C. HELDT, G. HUHN AND D. SCHOMBURG. 2004. BRENDA, the enzyme database: updates and major new developments. Nucleic Acids Res. **32**: D431–3.

112 SCHUSTER, R. AND H. G. HOLZHÜTTER. 1995. Use of mathematical models for predicting the metabolic effect of large-scale enzyme activity alterations. Application to enzyme deficiencies of red blood cells. Eur. J. Biochem. **229**: 403–18.

113 SCHUSTER, R., H. G. HOLZHÜTTER AND G. JACOBASCH. 1988. Interrelations between glycolysis and the hexose monophosphate shunt in erythrocytes as studied on the basis of a mathematical model. Biosystems **22**: 19–36.

114 SCHUSTER, R., G. JACOBASCH AND H. G. HOLZHÜTTER. 1989. Mathematical modelling of metabolic pathways affected by an enzyme deficiency. Energy and redox metabolism of glucose-6-phosphate-dehydrogenase-deficient erythrocytes. Eur. J. Biochem. **182**: 605–12.

115 SCHUSTER, S. 2004. Metabolic pathway analysis in biotechnology. In KHOLODENKO, B. N. AND H. V. WESTERHOFF (eds.), *Metabolic Engineering in the Post Genomic Era*. Horizon Scientific, Wymondham: 181–208.

116 SCHUSTER, S. AND C. HILGETAG. 1994. On elementary flux modes in biochemical reaction systems at steady state. J. Biol. Syst. **2**: 165–82.

117 SCHUSTER, S. AND T. HÖFER. 1991. Determining all extreme semi-positive conservation relations in chemical reaction systems. A test criterion for conservativity. J. Chem. Soc. Faraday Trans. **87**: 2561–6.

118 SCHUSTER, S., D. A. FELL, T. PFEIFFER, T. DANDEKAR AND P. BORK. 1998. Elementary modes analysis illustrated with human red cell metabolism. In LARSSON, C., I.-L. PÅHLMAN AND L. GUSTAFSSON (eds.), *BioThermoKinetics in the Post Genomic Era*. Chalmers, Göteborg: 332–9.

119 SCHUSTER, S., T. DANDEKAR AND D. A. FELL. 1999. Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. Trends Biotechnol. **17**: 53–60.

120 SCHUSTER, S., D. A. FELL AND T. DANDEKAR. 2000. A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. Nat. Biotechnol. **18**: 326–32.

121 SCHUSTER, S., C. HILGETAG, J. H. WOODS AND D. A. FELL. 2002. Reaction routes in biochemical reaction systems: Algebraic properties, validated calculation procedure and example from nucleotide metabolism. J. Math. Biol. **45**: 153–81.

122 SCHUSTER, S., M. MARHL AND T. HÖFER. 2002. Modelling of simple and complex calcium oscillations. From single-cell responses to intercellular signalling. Eur. J. Biochem. **269**: 1333–55.

123 SCHWENDER, J., F. GOFFMAN, J. B. OHLROGGE AND Y. SHACHAR-HILL. 2004. Rubisco without the Calvin cycle improves the carbon efficiency of developing green seeds. Nature **432**: 779–82.

124 SEGEL, I. 1993. *Enzyme Kinetics. Behavior and Analysis of Rapid Equilibrium and Steady-State Enzyme Systems*. Wiley, New York, NY.

125 SELKOV, E. E. 1968. Self-oscillations in glycolysis. 1. A simple kinetic model. Eur. J. Biochem. **4**: 79–86.

126 SELKOV, E. E. 1975. Stabilization of energy charge, generation of oscillations and multiple steady states in energy metabolism as a result of purely stoichiomteric regulation. Eur. J. Biochem. **59**: 151–7.

127 SELKOV E, S. BASMANOVA, T. GAASTERLAND, I. GORYANIN, Y. GRETCHKIN AND N. MALTSEV. 1996. The metabolic pathway collection from EMP: the enzymes and metabolic pathways database. Nucleic Acids Res. **24**: 26–8.

128 SEO, H., D.-Y. LEE, S. PARK, L. T. FAN, S. SHAFIE, B. BERTÓK AND F. FRIEDLER. 2001. Graph-theoretical identification of pathways for biochemical reactions. Biotechnol. Lett. **23**: 1551–7.

129 SERESSIOTIS, A. AND J. E. BAILEY. 1988. MPS: An artificially intelligent software system for the analysis and synthesis of metabolic pathways. Biotechnol. Bioeng. **31**: 587–602.

130 SIMONS, A., N. DAFNI, I. DOTAN, Y. ORON AND D. CANAANI. 2001. Establishment of a chemical synthetic lethality screen in cultured human cells. Genome Res. **11**: 266–73.

131 SIMPSON, T. W., B. D. FOLLSTAD AND G. STEPHANOPOULOS. 1999. Analysis of the pathway structure of metabolic networks. J. Biotechnol. **71**: 207–23.

132 SLEPCHENKO B. M., J. C. SCHAFF, I. MACARA AND L. M. LOEW. 2003. Quantitative cell biology with the Virtual Cell. Trends Cell Biol. **13**: 570–6.

133 SNOEP J. L., F. BRUGGEMAN, B. G. OLIVIER AND H. V. WESTERHOFF. 2006. Towards building the silicon cell: a modular approach. Biosystems **83**: 207–16.

134 SOMOGYI, R. AND J. W. STUCKI. 1991. Hormone-induced calcium oscillations in liver cells can be explained by a simple one pool model. J. Biol. Chem. **266**: 11068–77.

135 STELLING, J., S. KLAMT, K. BETTENBROCK, S. SCHUSTER AND E. D. GILLES. 2002. Metabolic network structure determines key aspects of functionality and regulation. Nature **420**: 190–3.

136 STEPHANOPOULOS, G. N., A. A. ARISTIDOU AND J. NIELSEN. 1998. *Metabolic Engineering: Principles and Methodologies.* Academic Press. San Diego, CA.

137 STROGATZ, S. H. 2001. *Nonlinear Dynamics and Chaos*. Perseus, Cambridge MA.

138 STRYER, L. 1995. *Biochemistry.* Freeman, NewYork, NY.

139 TAKAHASHI, K., N. ISHIKAWA, Y. SADAMOTO, et al. 2003. E-Cell 2: multi-platform E-Cell simulation system. Bioinformatics **19**: 1727–9.

140 TEUSINK, B., J. PASSARGE, C. A. REIJENGA, et al. 2000. Can yeast glycolysis be understood in terms of *in vitro* kinetics of the constituent enzymes? Testing biochemistry. Eur. J. Biochem. **267**: 5313–29.

141 TING, I. P. AND M. GIBBS. 1982. *Crassulacean Acid Metabolism.* American Society of Plant Physiologists, Rockville, MD.

142 TOMITA, M. 2001. Whole-cell simulation: a grand challenge of the 21st century. Trends Biotechnol. **19**: 205–10.

143 TONG, A. H., G. LESAGE, G. D. BADER, H. DING, H. XU AND X. XIN et al. 2004. Global mapping of the yeast genetic interaction network. Science **303**: 808–13.

144 URBANCZIK, R. AND C. WAGNER. 2005. An improved algorithm for stoichiometric network analysis: theory and applications. Bioinformatics **21**: 1203–10.

145 VARMA A. AND B. O. PALSSON. 1993. Metabolic capabilities of *Escherichia coli*. I. Synthesis of biosynthetic precursors and cofactors. J. Theor. Biol. **165**: 477–502.

146 VON KAMP, A. AND S. SCHUSTER. 2006. Metatool 5.0: fast and flexible elementary modes analysis. Bioinformatics.

147 WADDELL, T. G., P. REPOVIC, E. MELÉNDEZ-HEVIA, R. HEINRICH AND F. MONTERO. 1999. Optimization of glycolysis: new discussions. Biochem. Educ. **27**: 12–3.

148 WAGNER, A. AND D. A. FELL. 2001. The small world inside large metabolic networks. Proc. Roy. Soc. (Biol. Sci.) **268**: 1803–10.

149 WAGNER, C. 2004. Nullspace approach to determine elementary modes of chemical reaction systems. J. Phys. Chem. B **108**: 2425–31.

150 WATSON, M. R. 1986. A discrete model of bacterial metabolism. CABIOS **2**: 23–7.

151 WIBACK, S. J. AND B. O. PALSSON. 2002. Extreme pathway analysis of human red blood cell metabolism. Biophys. J. **83**: 808–18.

152 WIBACK, S. J., R. MAHADEVAN AND B. O. PALSSON. 2003. Reconstructing metabolic flux vectors from extreme pathways: defining the alpha-spectrum. J. Theor. Biol. **224**: 313–24.

153 WICK, L. M., M. QUADRONI AND T. EGLI. 2001. Short- and long-term changes in proteome composition and kinetic properties in a culture of *Escherichia coli* during transition from glucose-excess to glucose-limited growth conditions in continuous culture and vice versa. Environ. Microbiol **3**: 588–99

154 WILHELM, T., J. BEHRE AND S. SCHUSTER. 2004. Analysis of structural robustness of metabolic networks. IEE Syst. Biol. **1**: 114–20.

155 WINZELER, E. A., D. D. SHOEMAKER, A. ASTROMOFF, H. LIANG, K. ANDERSON AND B. ANDRE. 1999. Functional characterization of the *S. cerevisiae* genome by gene deletion and parallel analysis. Science **285**: 901–6.

156 WOLF, J. AND R. HEINRICH. 2000. The effect of cellular interaction on glycolytic oscillations in yeast: a theoretical investigation. Biochem. J. **345**: 321–34.

157 WU, L., W. WANG, W. A. VAN WINDEN, W. M. VAN GULIK AND J. J. HEIJNEN. 2004. A new framework for the estimation of control parameters in metabolic pathways using lin-log kinetics. Eur. J. Biochem. **271**: 3348–59.

158 ZASLAVER, A., A. E. MAYO, R. ROSENBERG, P. BASHKIN, H. SBERRO, M. TSALYUK, M. G. SURETTE AND U. ALON. 2004. Just-in-time transcription program in metabolic pathways. Nat. Genet. **36**: 486–91.

21 Inferring Gene Regulatory Networks

Michael Q. Zhang

1 Introduction

Gene expression (normally referring to the cellular processes that lead to protein production) is responsive to environmental signals, and is controlled and regulated at multiple levels [108]. In addition to protein-coding genes, there are also non (protein)-coding RNA (ncRNA) genes, such as tRNAs, rRNAs, small nucleolar RNAs (snoRNAs), small interfering RNAs (siRNAs)/micro RNAs (miRNAs), antisense RNAs and many others [74]. A molecular network is often represented by a graph of *nodes* (representing molecules) and *links* (representing interactive relations between pairs of molecules). Such interactions can be either symmetric, indicated by an undirected link, or asymmetric, indicated by a directed link. There are many definitions (or incarnations) of gene regulatory networks (GRNs) depending on the nature of the nodes (i.e. DNAs, RNAs, proteins or a mixture) and the links (i.e. regulatory relations). For example, signaling networks (one type or one part of GRNs that is discussed in more detail in Chapter 22) are protein networks that describe post-translational modification, degradation, subcellular localization or physical interaction of proteins. In this chapter, the discussion is mainly focused on protein–DNA and protein–RNA interactions, as such so-called *trans*-factor and *cis*-element interactions are the key for understanding how eventually genes are turned on or off. Therefore, knowledge of these interactions is a prerequisite for building any gene regulation pathway or more extended networks. The central question is how various input signals are integrated at each step of gene expression to produce differential output of the gene products in order to respond to different physiological conditions in different types of cells at different developmental stages.

Bioinformatics - From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

2 Gene Regulation at the Transcriptional Level

Transcription is the first and the major step in gene expression during which genetic information is transcribed from DNA to RNA transcript. Molecularly, there are several substeps (e.g. initiation of the transcription, promoter escape, whereby the polymerase overcomes the pausing at nucleosomes, elongation of the transcript and termination of transcription) each of which can be subject to regulation. Transcription factors (TFs) binding to specific promoter DNA (*cis*-elements) is the most basic requirement for recruiting the transcription machinery (polymerase and associated factors) to form the preinitiation complex (PIC) at the transcriptional start site (TSS), and for switching on and off RNA transcript synthesis. Co(transcription) factors (CTFs) that do not directly bind to DNA, but to other molecules of the PIC, are also crucial as they can help activating or deactivating TFs or transcription machinery and can modify histones or chromatin structures (see Chapters 6 and 45, and reviews, e.g. Ref. [132]).

Although many mathematical models have been developed for modeling single-cell dynamics of highly simplified systems ranging from lactose metabolism in bacteria [118] to cell cycle control in yeast [26, 97], they are largely based on known factors, pathways and network topologies. Given the complexity of the GRNs and very limited dynamical measurements with current experimental technologies, it may be still too early to build comprehensive and accurate dynamical models for truly realistic GRNs. (It is a fascinating problem why a biological system is robust against fluctuating environments, for recent review on the stochasticity problem in realistic dynamic modeling, see Ref. [81].) The majority of the new computational methods for detecting *cis*–*trans* relationships are based on modern statistical or machine learning approaches; these methods are quite effective in the discovery of *cis*-regulatory elements (single motifs) or *cis*-regulatory modules (CRMs, also called motif combinations or composite motifs) and the inference of GRN modules. A GRN module is a triple of TFs/CTFs, *cis*-element and target genes).

In the study of transcriptional regulation, there are two related problems: one is to identify regulatory regions, i.e. promoters or TSSs (they may also include introns and 3′ flanking regions in vertebrates); another is to detect *cis*-element motifs [most of them are TF-binding sites (TFBSs)]. In Chapter 6, Werner has given an excellent description of *de novo* promoter prediction methods, and the interplay between promoter recognition and *cis*-element analysis. (One can also find related reviews, e.g. in Refs. [165–167]). In this chapter, I focus on finding *cis*-element motifs or network modules using large-scale functional genomics microarray data.

2.1 Finding TFBSs and Motifs

The traditional approach for finding *cis*-elements is to collect a set of (target gene) promoter sequences believed to be enriched by some common TFBS motifs. They may either be collected from the literature or from systematic experiments [such as SELEX (selection-amplification), etc.]. There are many *de novo* TFBS motif-finding algorithms that may be used. For recent reviews on computational TFBS finding methods, see, e.g. Refs. [19, 120, 126, 156, 166], and see Ref. [153] for a recent benchmark of some popular motif finders. In addition to the three classical alignment-based motif-finding algorithms, i.e. CONSENSUS [64], MEME [3] and Gibbs motif sampler [115], most modern approaches have tried to extend either to discover motif combinations or CRMs (e.g. Refs. [52, 53, 57]), or to use evolutionary conservation or comparative genomic information. The latter approach is also called phylogenetic foot printing, e.g. Refs. [106, 113, 138, 155, 162]. There are also combinations of both approaches, e.g. Refs. [122, 151, 161]. One can also increase specificity by incorporating structural information [84, 107], e.g. if the protein binds as a homodimer, one could restrict the search to palindromic motifs.

More powerful and flexible motif finders can take advantage of a separate sequence set called the *background set*, serving as a negative control. The goal is to search only for motifs that are most discriminating, i.e. only those enriched in the foreground set relative to the background set. Such motif finders are called *discriminant motif finders*. Examples are ANN-Spec [159], PSSM [6], DMOTIFS [137], DWE [146] and DME [141].

2.2 Identifying Target Genes

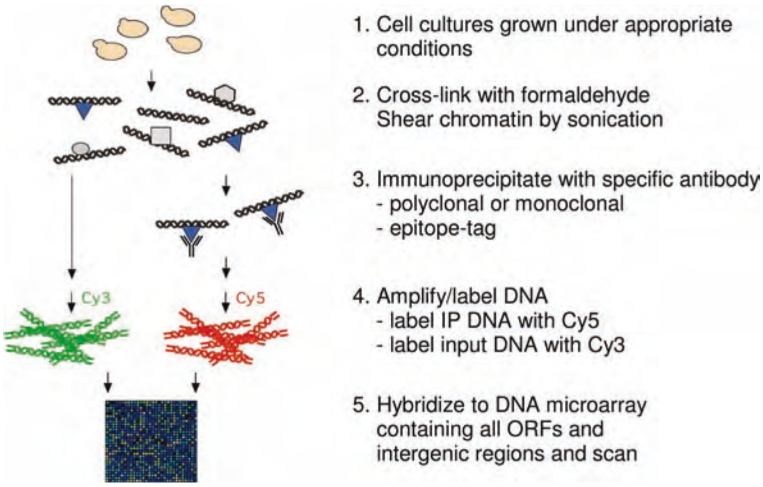
For known TFs, one needs either to get or to construct their nucleotide weight matrices (NWMs). (For a definition and discussion of NWMs, see Chapter 6.) Currently there are at least two new types of microarrays for measuring *in vitro* protein–DNA binding –protein binding microarrays (PBMs [114]) and DIP-chip (DNA immunoprecipitation with microarray detection [103]). PBM uses purified and tagged proteins binding to an array of double-stranded DNA oligos. In DIP-chip, protein–DNA complexes are immunoprecipitated from an *in vitro* mixture of purified protein and naked genomic DNA fragments and the sequences of the bound DNA fragments are determined by hybridization to a genomic DNA array. Both methods are very useful for systematically studying TF-binding specificity and hence defining motif-scoring matrices, because most motifs in the existing transcription factor databases, such as SCPD (yeast promoter database [171]), TRANSFAC (general commercial transcription factor database [63]), JASPAR (derived database from TRANFAC [133]) and TRED (cancer-related transcriptional regulation element database [169]), may still be very limited and therefore biased. It would only be possible to use known motif matrices to search for target genes computationally when the expectation of finding such a motif is much smaller than in a random promoter region. This means (i) the motif should be long and/or highly conserved, (ii) multiple motifs form clusters (homotypic or heterotypic modules) or (iii) the search space could be further restricted (to conserved regions, to nonrepeat regions, to be close to TSS, etc.) so that the total information content is well above the background noises.

2.3 Discovering Novel Motifs and Target Genes

Often, neither the motif nor target genes are known. Then motif finding and target identification have to be done simultaneously and/or iteratively. Comparative sequence analysis by itself may give some reasonable results for simple organisms [125]; however, in general, microarray data are required for any comprehensive studies. Currently, two major types of microarray data have provided large-scale input for transcriptional GRN studies. One is mRNA expression microarray data (see Chapter 24) and the other is localization data obtained by the ChIP-chip method (also called ChIP-on-chip, see Refs. [70, 127] and Chapter 45). ChIP-chip differs from DIP-chip by formaldehyde cross-linking proteins to chromatin DNA *in vivo*. Instead of using hybridization with genomic DNA arrays as in the ChIP-chip method, the SACO (serial analysis of chromatin occupancy [76]) method employs SAGE-tag sequencing after chromatin immunoprecipitation.

There are many approaches based on expression microarrays. The expression data is the direct readout of mRNA responses for a given condition; it becomes much more powerful when data is collected under multiple conditions or at multiple time points. The biggest problem in practice is that it is difficult to identify the direct targets after a treatment or perturbation to the cells. The best data for motif/target identification can be obtained when a perturbation is done directly on DNA-binding TFs (such as by TF knockout, mutation or knockdown, see, e.g. Ref. [25]). Things become more difficult when perturbation is more upstream, which often causes the activation of many pathways, as in heat shock or other stress responses, or when the top of a signal transduction pathway is perturbed, as it happens in drug treatment at the receptor level. As multiple TFs are likely to be involved in such general perturbations, responsive target genes would consist of a mixture of direct targets for different TFs. Another complication arises if the sample is heterogeneous, either due to technical difficulties of pure sample extraction or due to intrinsic mixtures such as embryonic stem cell differentiation. Classical approaches were based on clustering analysis in order to identify correlated gene expression patterns from which one could try to identify coregulated genes and common *cis*-elements enriched in their promoters (e.g. Ref. [72, 142]). When microarray data from a large number of conditions are combined, more sophisticated two-dimensional or biclustering methods must be used to identify significant expression patterns involving only some subset of genes and subset of conditions [21, 28, 47]. For more information on clustering of expression microarrays, see also Chapters 24, 26 and 27, and Ref. [56]. Since coexpressed genes may not all be coregulated, and genes expressed at different conditions and times may be regulated by the same TF, other information has also been used for filtering for functionally coregulated genes, such as, using Gene Ontology (GO) annotations, protein–protein interactions, metabolic networks or localization/binding data, etc., or combining potential target genes at different time-points by using TF expression profiles (e.g. Refs. [14, 172]. Since clustering can help in motif identification and good motifs can help refining clusters, MDscan [104] tries to iterate the process in order to sample the "good" genes and to weed out the "bad" ones. Models using joint likelihoods for sequence and expression have also been developed [7, 68].

Finding discriminating motifs that can best classify the foreground promoters of the responsive genes from the background promoters of the nonresponsive genes may be used for motif discovery with expression microarray data. The most powerful generalization of this idea would be to turn motif finding into a feature selection problem in regression analysis by asking what is the set of features X (some functions of the motifs or CRMs) that can best explain the expression data Y (e.g. as scored via the log-ratio of the expression data under a given condition). This is very similar to the general problem in genetics: Y represents the phenotype (mRNA expression) and X represents the genotype (measures of the DNA elements). One would like to learn a model (function f) so that $f(X)$ can best predict Y. When "best" is measured by the average squared error based on the distribution $Pr(X,Y)$, the solution is the conditional expectation (also known as the regression function, see, e.g. Ref. [61]: $f(X) = E(Y \mid X = x)$. REDUCE is the first successful motif-selection algorithm based on linear regression [20]. It has been generalized later to include cross-interaction terms [85], to use NWMs discovered by MDscan (Motif_Regressor [31]), to apply logistic regression [86] and MARS (multivariate adaptive regression splines [43]) – a nonlinear model based on regression trees (MARS_Motif [33]). Using a simple Bayesian network to learn the AND, OR and NOT logic with constraints on motif strength, orientation and relative position from a training set of yeast cycle cell genes and their expression patterns (coded as binary on–off variables), Beer and Tavazoie [11] demonstrated that the inferred regulatory rules can correctly predict expression patterns for about 70% of other cell cycle genes (not used for training the model) in the same arrays.



1. Cell cultures grown under appropriate conditions
2. Cross-link with formaldehyde
   Shear chromatin by sonication
3. Immunoprecipitate with specific antibody
   - polyclonal or monoclonal
   - epitope-tag
4. Amplify/label DNA
   - label IP DNA with Cy5
   - label input DNA with Cy3
5. Hybridize to DNA microarray containing all ORFs and intergenic regions and scan

Figure 1 ChIP-chip example: genome-wide localization analysis in yeast [128].

Almost all the tools developed for analyzing expression microarray data can also be easily applied to the analysis of localization data, e.g. of protein-binding sites in the genome, such as ChIP-chip data. Although ChIP-chip is a global measurement for *in vivo* binding of proteins to chromatin DNA and, hence, is potentially capable of revealing direct target genes – most targets identified in expression arrays are not direct targets – due to the current resolution and to non-specific or non-functional cross-links, not all putative targets are functional or possess functional *cis*-elements. ChIP-chip data have also been used to further refine motifs found by expression data (e.g. using boosting approach [69]). The most comprehensive ChIP-chip analysis done to date is the study of 203 TFs in yeast (Figure 1); since experimental resolution can only narrow down the binding sites to 200- to 1000-kb regions, further computational analysis is required for functional motif discovery. Using genome alignment information of four related yeast species and six *de novo* motif finders, a transcriptional regulatory code for the yeast genome has been inferred [58].

2.4 Inferring GRN Modules and Integrating Diverse Types of Data

Fully probabilistic graphical models, Bayesian networks in particular, have become very popular in GRN module inference. They are flexible with respect to the integration of diverse data sets (see also Chapter 35; for a recent review, see Friedman [44]). Bayesian networks have frequently been used to infer regulatory relationships between a regulator and its (often indirect) target given the expression data, ChIP-chip data were initially used as an informative prior to bias the Bayesian network model (posterior probability) towards potential direct regulator–target pairs [13,60] or as a noisy sensor [135]. If a parent gene in the hierarchical Bayesian network is a TF, its children may share its binding side motif in their promoter DNA. This idea may be used to refine Bayesian networks iteratively [149].

To reconstruct robust GRNs, network modules of genes that are coregulated under a set of experimental conditions have been proposed [75, 136] in which all the genes within a single module are controlled by a common regulatory program (shared set of TFs/*cis*-elements). When ChIP-chip data is also incorporated, physical links between TFs and their targets allow for building more robust GRNs (GRAM [5]). Inferring modules simultaneously using all three data types, i.e. Motif, ChIP-chip and expression microarray, have only been tried with the yeast cell cycle data. Here, a two-step process was used. The seed construction step predicts the putative modules consisting of TFs, their binding motifs and the common expression profile; the validation step filters false positives, and determines the module size and function [36]. Similar results have also been obtained previously by an iterative procedure for reconstruction of the yeast cell cycle regulation network [83] (see Banerjee and Zhang [4] for a specific review on yeast cell cycle regulatory networks).

Expression microarray data can also been regarded as quantitative phenotypes for so-called genetic genomics analysis. In such an approach, the genetic mechanisms of segregation and recombination are used to reshuffle the genomes of two or more donor parents, to produce a population of segregating offspring with many combinations of gene variants, both *cis*-linked or *trans*-linked quantitative trait loci (QTL) may be mapped by comparing differential expression (the method is also called eQTL) in the segregating population [16,29]. The power has been further demonstrated recently with novel application in epistasis analysis for determining the order of function of genes in pathways [154]. Although differential mRNA expression levels are used, the regulatory relation inferred by eQTL analysis can be at any level: from the genomic (*cis*–*trans*) level to the protein (interaction) level.

3 Gene Regulation at the Level of RNA Processing

The next level of gene regulation is at the level of RNA processing, which includes capping, splicing, polyadenylation, editing, degradation (including nonsense mediated decay) and transport; many of these (in particular, the first three) steps are co-transcriptional and hence directly coupled to transcription [108]. Hence, the steady state of the transcript level measured in an expression microarray is the result of a combined (or net) effect (see Grigull [51] for a microarray RNA stability study after a general transcriptional shutoff). In this section, we will only focus on RNA splicing, especially alternative splicing that is responsible for generating diverse protein isoforms from a single gene locus. Alternative splicing plays many critical roles in regulatory pathways in metazoans, including those controlling cell growth, cell death, differentiation and development, and its misregulation has been implicated in a large number of human diseases [143]. Genomic technologies for RNA-splicing regulation studies are much less developed, largely due to the difficulty of handling RNAs and to much more complex structure and dynamics of the spliceosome, which incorporates more than 300 protein/RNA components and thus has significantly more parts then the transcriptional (e.g. the PIC) or translational (ribosome) apparatus. Computational prediction of alternative exons is still in its infancy – almost all the exon–intron annotations are primarily based on cDNAs and expressed sequence tags (ESTs) [93,166].

3.1 Identification of Splicing Enhancers and Silencers

Like the TFs in transcription, splicing factors (SFs) play important roles in regulating RNA splicing. Most well-characterized SFs belong to one of two classes: heterogeneous nuclear ribonucleoproteins (hnRNPs) and arginine/serine-rich (SR) proteins, besides some TFs and elongation factors (see reviews, e.g. Refs. [15, 170]). There have been only a few large-scale discovery and characterization projects of splicing enhancers – mostly bound by SR proteins in exons – using SELEX through either a binding assay [148] or a functional assay [32, 101, 102]. Even less data is available on intron elements, many of which are silencers often bound by hnRNPs [18,73] (see review by Ladd and Cooper [90]). Hence there are limited entries in SF-binding site databases [144, 145, 150]. Several computational approaches have also been developed for finding putative *cis*-elements genome-wide *in silico* [22,41,42,124,139,168].

3.2 Splicing Microarrays

The feasibility of using microarrays to study RNA splicing regulation was first demonstrated in yeast [30]. Since 40–60% of mammalian genes have introns (a typical gene has about eight introns, see also Chapter 45), compared to 3.8% intron-containing genes in yeast, detecting alternative splicing in a mammalian system with microarrays has only become possible recently [79, 82,98,119,163]. Often, genes have many complex alternative-splicing isoforms that cannot be tracked unambiguously by the limited number of exon or junction probes – the biggest challenge of all splicing array experiments is data analysis and biological interpretation (see review, e.g. Ref. [92]). As transcription and splicing are intrinsically coupled, it may not be possible to separate their individual contributions to the steady-state levels of the transcripts despite some initial attempts [91, 97, 164]. It is interesting that transcription and alternative splicing appear to act independently on different sets of genes in order to define tissue-specific expression profiles [119].

4 Gene Regulation at the Translational Level

Regulated translation controls a wide range of cellular processes in eukaryotes. Like regulation at the other gene expression levels, global (cell-wide) regulation can be exercised through modification of the basic translation machinery via events such as phosphorylation. Otherwise, control is more selective with sequence-specific RNA-binding proteins recognizing target transcripts, thereby regulating translation. While the number of such factors is growing rapidly, the molecular details of how they regulate translation are not well understood for most of them. The binding sites for many of these regulatory proteins are located in 5′- or 3′-untranslated regions (UTRs) of the target transcript [37, 67, 123, 130, 158]. The databases UTRdb and UTRsite maintain a collection of sequence and regulatory motifs in the UTRs of eukaryotic mRNAs [112]. Computational studies of sequence elements in UTRs have been used mainly for UTR classification [35] or prediction of start or polyadenylation sites [49, 54, 94, 147]. Comparative genomics has also been applied to detect conserved mammalian *cis*-regulatory elements in 3′-UTRs [162]. One of the best studied *cis*-regulatory motifs in the 5′-UTR is the internal ribosome entry site (IRES) [88] and the one in the 3′-UTR is the PUF/pumilo binding site [46, 157]. SELEX has also been applied to studying UTR motifs [71]. The PUF family of RNA-binding proteins is highly conserved and plays important roles in regulating mRNA transcript satiability. The first human PUF homolog – PUM2 target: P2P-R – has also been identified by a microarray study using WebQTL (similar to the eQTL mentioned above [134]). Another famous example of translational regulators is the nuclear–cytoplasmic shuttling protein HuR. It has the dual functions of regulating RNA stability through the AU-rich element in its 3′-UTR and differentially repressing cap-dependent and IRES-mediated translational initiation via 5′-UTR binding, and its targets and binding sites have also proved to be amenable to genome-wide studies [111]. In addition to expression microarrays, high-throughput genomic arrays are currently being developed and tested for large-scale identification of *cis*-regulatory elements related to RNA binding proteins. This technology is similar to ChIP-chip for detecting DNA-binding targets and is termed ribonomics [46, 121]. Genome-wide translational regulatory network studies are clearly on the horizon. A comprehensive review on UTR elements and post-transcriptional regulation in cancer may be found in [1].

5 Gene Regulation by Small ncRNAs

RNA transcripts that are not mRNA, tRNA or rRNA are often referred to as ncRNAs. They include small nuclear RNA (snRNA), snoRNA, small temporal RNA (stRNA), Xist (X inactive specific transcripts)-like RNA, antisense RNA, processed pseudogenes, short RNAs (miRNA), siRNA, trophoblast ncRNA (tncRNA), small modulatory RNA (smRNA) [17] and other RNA transcripts of unknown function (TUFs) [27]). Here, we only focus on miRNA (some results are also applied to other short RNAs, such as siRNA) discovered recently that can add "micromanagement" fine-tuning layers in complex GRNs [8, 9]. Processed by the RNA interference machinery, these short (around 22 bp) RNAs can lead to sequence-specific inhibition of gene expression either at the transcriptional level – through chromatin modification and epigenetic silencing – in the nucleus or at the post-transcriptional level – through target message cleavage or degradation, or alternatively through translational repression – in the cytoplasm [62, 110, 152]. Recently, microarrays have been used for profiling tissue-specific expression of miRNAs [2, 100] as well as for identifying target genes [99]. It has been found that miRNAs clustered within 50 kb tend to be cistronic (transcribed together as a single transcript) and those in introns tend to transcribe together with the host mRNA [10].

Based on training samples in databases, such as the microRNA Registry (which contains 227 human miRNAs in release 6.0) and in Rfam [50], there has been a tremendous flood of computational programs and predictions for miRNAs (based on conservation, secondary structure, etc.) as well as their target genes (based on complementarity, conservation, etc.), mostly in the 3′-UTR (see the recent reviews of Bengert and Dandekar [12] and Brown and Sanseau [17]). In animals, imperfect complementarity with miRNAs makes computational prediction of their targets very difficult. Although it has been computationally predicted that there may be about 2000 [78], 5000 [162] or more than 5300 [96] human miRNA targets, given that we do not yet know even a single authentic training sample (an experimentally characterized human endogenous target gene *in vivo*), we must be very cautious about those computational predicted targets. In a zebrafish maternal-zygotic Dicer null, where all miRNAs are globally removed, many of the early developmental pathways computationally predicted to be targeted by miRNAs remain intact [48]. This experimental result implies that either the predicted targets are wrong or they have no functional consequences. The miRNA target gene regulatory code mostly in 3′-UTRs appears to be very similar to the transcriptional *cis*-regulatory code in promoters. Specificity mainly comes from combinatorial binding of multiple regulators, as each individual miRNA can target multiple genes [66]. With tissue-specific microarray data and motif-finding algorithms, identification of functional target genes and *cis*-regulatory sites may be more reliable [99]. Incorporation of known secondary structure information can further improve the target prediction accuracy [131]. Using a conservation filter (which require target recognition sites must be conserved among ortholog genes) of eight vertebrate species, a new algorithm PicTar [89] is able to predict tissue-specific mammalian target genes for a combination of miRNAs. This method was used to estimate about 200 targets per miRNA in the mammals. After all, miRNAs and TFs form an integrated regulation network – natural feedback loops: pre-miRNAs are regulated by TFs and many of the miRNA target themselves are TFs. It is a computational challenge to be able to treat them simultaneously. It has been demonstrated experimentally that such fascinating networks can regulate left and right symmetry in nematode chemosensory neuron localizations, and the two miRNAs that repress the die-1 zinc-finger TF were totally missed by previous computational predictions [24]. Another endogenous feedback loop was recently uncovered in the mammalian cell cycle regulation system where two miRNAs regulated by c-Myc modulate another c-Myc activated target E2F1 [116].

6 GRNs in Development and Evolution

The best context in which to study GRNs is metazoan organ development (see the special section in *Proceedings of the National Academy of Sciences USA* 2005, vol. 102, no. 14). Unlike the dynamics at the single-cell level, tissue lineage differentiation and organogenesis involve much slower dynamics of creation and evolution of cellular populations. In the terminally differentiated tissues, the regulatory program is often relatively simple, *cis*-regulatory regions in promoters tend to be closer to TSS and many TFBS are less conserved among distant species. In fact, it has been shown recently that proximal promoters can predict tissue-specific gene expression [140]. In contrast, the regulatory programs that control early development tend to be much more complex, almost always involving distal enhancers and/or complicated locus control regions (LCRs). However, lineage developmental master TFs and their binding sites (hence the core networks) are often more conserved during evolution. Using conservation and structural properties of TF modules (inter-distance and orientation), large-scale computational identification of distal enhancers has become possible [55]. Developmental GRNs refer to “logic maps of the control functions that direct development, and they relate these maps directly to the genomic regulatory sequence” [117]. They are typically large scale, multilayered, and organized in a nested, modular hierarchy of regulatory network kernels, function-specific building blocks and structural gene batteries. They are also inherently multicellular and involve changing topological relationships among a growing number of cells [105]. The two best-studied early developmental systems are the specification of the endomesoderm in sea urchin embryos and the dorsal–ventral patterning in the *Drosophila* embryo [95]. The central problem for GRN reconstruction is to identify CRMs and to figure out what kind of logic function, i.e. mapping from regulatory signal input to protein production output, each DNA module is programmed to compute and how different modules are integrated in the circuit [77].

Since the regulation of gene expression is ultimately the result of the evolutionary response to the challenge of surviving in a changing environment in the past history [23], we need to understand what is the core (conserved) subnetwork and what are the species-specific innovations. Different species may use different pathways to accomplish the same task or may invent unique pathways for their survival in a particular niche or environment. Evolutionary comparison of GRNs of related species is essential [38,160] (also Ref. [34] for a discussion on evolution-development convergence). Studies on conservation and evolution of *cis*-regulatory architectures [45], of global expression profiles [39, 80, 109] or of GRNs [40, 65] have yielded valuable insight on history, diversity and function of various genetic regulatory systems.

The obvious future task is to develop high-throughput experimental technologies to measure dynamic gene regulation at many different levels simultaneously and to develop large-scale computational tools to integrate diverse data (DNA, RNA, protein, metabolites, etc.) in order to obtain a coherent picture at the systems biology level [59,105]. The current emphasis should not be on graphs or "regulatory linkage, but the *nature of biological systems that allows gene products to be linked together* in many nonlethal and even useful combinations" [87]. Such a comprehensive study is important for inferring regulatory programs not only in normal biological systems, but also in pathological systems, such as in cancer [129].

The central dogma of molecular biology, stating that the flow of genetic information from DNA to RNA to protein is hardly an archaic myth. Yet, the subsequent discoveries that RNA can be transcribed back to DNA ("reverse transcription"), that it can cause gene silencing ("epigenetic control" or imprinting) as well as degradation of mRNA transcripts or repress translation ("RNA interference") and that it can also function enzymatically like a protein (ribozymes, cleave messages) have forced people to rethink: what is a gene and in how many ways can it be regulated? The fact that information may flow in more than one direction compels us to wonder if RNA may be the truly central player with both the informational capacity of DNA and the functional capacity of protein? With the dual role of information carrier and processor, RNA cannot only store information into DNA for more stable preservation of information (genetic structure), but also translate information into protein for more flexible and efficient processing of information (biochemical function).

Acknowledgments

M.Q.Z.s laboratory is supported by grants from NIH/NCI, NSF and CSHL Associations.

References

1 Audic, Y. and R. S. Hartley. 2004. Post-transcriptional regulation in cancer. Biol. Cell **96**: 479–98.

2 Babak, T., W. Zhang, Q. Morris, B. J. Blencowe and T. R. Hughes. 2004. Probing microRNAs with microarrays: tissue specificity and functional inference. RNA **10**: 1813–9.

3 Bailey, T. L. and C. Elkan. 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proc. ISMB **2**: 28–36.

4 Banerjee, N. and M. Q. Zhang. 2005. Transcription regulatory networks in yeast cell cycle. In Shannon, F. and S. Rao (eds.), *Microarray Expression Profiling and Transcriptional Regulatory Networks*, Landes Bioscience / Eureka Publishing, Austin, TX.

5 Bar-Joseph, Z., G. K. Gerber, T. I. Lee, et al. 2003. Computational discovery of gene modules and regulatory networks. Nat. Biotechnol. **21**: 1337–42.

6 Barash, Y., G. Bejerano and N. Friedman. 2001. A simple hyper-geometric approach for discovering putative transcription factor binding sites. Presented at the First International Workshop on Algorithms in Bioinformatics. Lecture Notes Comput. Sci. **2149**: 278–93.

7 Barash, Y. and N. Friedman. 2002. Context-specific Bayesian clustering for gene expression data. J. Comput. Biol. **9**: 169–91.

8 Bartel, D. P. 2004. MicroRNAs: genomics, biogenesis, mechanism, and function. Cell **116**: 281–97.

9 Bartel, D. P. and C. Z. Chen. 2004. Micromanagers of gene expression: the potentially widespread influence of metazoan microRNAs. Nat. Rev. Genet. **5**: 396–400.

10 Baskerville, S. and D. P. Bartel. 2005. Microarray profiling of microRNAs reveals frequent coexpression with neighboring miRNAs and host genes. RNA **11**: 241–7.

11 Beer, M. A. and S. Tavazoie. 2004. Predicting gene expression from sequence. Cell **117**: 185–98.

12 Bengert, P. and T. Dandekar. 2005. Current efforts in the analysis of RNAi and RNAi target genes. Brief Bioinform. **6**: 72–85.

13 Bernard, A. and A. J. Hartemink. 2005. Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. Pac. Symp. Biocomput.: 459–70.

14 Birnbaum, K., P. N. Benfey and D. E. Shasha. 2001. *Cis* element/transcription factor analysis (*cis*/TF): a method for discovering transcription factor/*cis* element relationships. Genome Res. **11**: 1567–73.

15 Black, D. L. 2003. Mechanisms of alternative pre-messenger RNA splicing. Annu. Rev. Biochem. **72**: 291–336.

16 Brem, R. B., G. Yvert, R. Clinton and L. Kruglyak. 2002. Genetic dissection of transcriptional regulation in budding yeast. Science **296**: 752–5.

17 Brown, J. R. and P. Sanseau. 2005. A computational view of microRNAs and their targets. Drug Discov. Today **10**: 595–601.

18 Buckanovich, R. J. and R. B. Darnell. 1997. The neuronal RNA binding protein Nova-1 recognizes specific RNA targets *in vitro* and *in vivo*. Mol. Cell. Biol. **17**: 3194–201.

19 BULYK, M. L. 2003. Computational prediction of transcription-factor binding site locations. Genome Biol. **5**: 201.

20 BUSSEMAKER, H. J., H. LI AND E. D. SIGGIA. 2001. Regulatory element detection using correlation with expression. Nat. Genet. **27**: 167–71.

21 CALIFANO, A., G. STOLOVITZKY AND Y. TU. 2000. Analysis of gene expression microarrays for phenotype classification. Proc. ISMB **8**: 75–85.

22 CARTEGNI, L., J. WANG, Z. ZHU, M. Q. ZHANG AND A. R. KRAINER. 2003. ESEfinder: a web resource to identify exonic splicing enhancers. Nucleic Acids Res. **31**: 3568–71.

23 CASES, I. AND V. DE LORENZO. 2005. Promoters in the environment: transcriptional regulation in its natural context. Nat. Rev. Microbiol. **3**: 105–18.

24 CHANG, S., R. J. JOHNSTON, JR., C. FROKJAER-JENSEN, S. LOCKERY AND O. HOBERT. 2004. MicroRNAs act sequentially and asymmetrically to control chemosensory laterality in the nematode. Nature **430**: 785–9.

25 CHEN, G., N. HATA AND M. Q. ZHANG. 2004. Transcription factor binding element detection using functional clustering of mutant expression data. Nucleic Acids Res. **32**: 2362–71.

26 CHEN, K. C., L. CALZONE, A. CSIKASZ-NAGY, F. R. CROSS, B. NOVAK AND J. J. TYSON. 2004. Integrative analysis of cell cycle control in budding yeast. Mol. Biol. Cell **15**: 3841–62.

27 CHENG, J., P. KAPRANOV, J. DRENKOW, et al. 2005. Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. Science **308**: 1149–54.

28 CHENG, Y. AND G. M. CHURCH. 2000. Biclustering of expression data. Proc. ISMB **8**: 93–103.

29 CHEUNG, V. G. AND R. S. SPIELMAN. 2002. The genetics of variation in gene expression. Nat. Genet. **32 (Suppl.)**: 522–5.

30 CLARK, T. A., C. W. SUGNET AND M. ARES, JR. 2002. Genomewide analysis of mRNA processing in yeast using splicing-specific microarrays. Science **296**: 907–10.

31 CONLON, E. M., X. S. LIU, J. D. LIEB AND J. S. LIU. 2003. Integrating regulatory motif discovery and genome-wide expression analysis. Proc. Natl Acad. Sci. USA **100**: 3339–44.

32 COULTER, L. R., M. A. LANDREE AND T. A. COOPER. 1997. Identification of a new class of exonic splicing enhancers by *in vivo* selection. Mol. Cell. Biol. **17**: 2143–50.

33 DAS, D., N. BANERJEE AND M. Q. ZHANG. 2004. Interacting models of cooperative gene regulation. Proc. Natl Acad. Sci. USA **101**: 16234–9.

34 DAVIDSON, E. H. 2001. *Genomic Regulatory Systems: Development and Evolution*. Academic Press, New York, NY.

35 DAVULURI, R. V., Y. SUZUKI, S. SUGANO AND M. Q. ZHANG. 2000. CART classification of human 5′ UTR sequences. Genome Res. **10**: 1807–16.

36 DE BIE, T., P. MONSIEURS, K. ENGELEN, B. DE MOOR, N. CRISTIANINI AND K. MARCHAL. 2005. Discovering transcriptional modules from motif, ChIP-chip and microarray data. Pac. Symp. Biocomput.: 483–94.

37 DE MOOR, C. H., H. MEIJER AND S. LISSENDEN. 2005. Mechanisms of translational control by the 3′ UTR in development and differentiation. Semin. Cell. Dev. Biol. **16**: 49–58.

38 DICKMEIS, T. AND F. MULLER. 2005. The identification and functional characterisation of conserved regulatory elements in developmental genes. Brief Funct. Genomic Proteomic **3**: 332–50.

39 ENARD, W., P. KHAITOVICH, J. KLOSE, et al. 2002. Intra- and interspecific variation in primate gene expression patterns. Science **296**: 340–3.

40 EVANGELISTI, A. M. AND A. WAGNER. 2004. Molecular evolution in the yeast transcriptional regulation network. J. Exp. Zool. B Mol. Dev. Evol. **302**: 392–411.

41 FAIRBROTHER, W. G., R. F. YEH, P. A. SHARP AND C. B. BURGE. 2002. Predictive identification of exonic splicing enhancers in human genes. Science **297**: 1007–13.

42 FAIRBROTHER, W. G., G. W. YEO, R. YEH, P. GOLDSTEIN, M. MAWSON, P. A.

Sharp and C. B. Burge. 2004. RESCUE-ESE identifies candidate exonic splicing enhancers in vertebrate exons. Nucleic Acids Res. **32**: W187–90.

43 Friedman, J. 1991. Multivariate adaptive regression splines. Ann. Stat. **19**: 1–141.

44 Friedman, N. 2004. Inferring cellular networks using probabilistic graphical models. Science **303**: 799–805.

45 Gasch, A. P., A. M. Moses, D. Y. Chiang, H. B. Fraser, M. Berardini and M. B. Eisen. 2004. Conservation and evolution of *cis*-regulatory systems in ascomycete fungi. PLoS Biol. **2**: e398.

46 Gerber, A. P., D. Herschlag and P. O. Brown. 2004. Extensive association of functionally and cytotopically related mRNAs with Puf family RNA-binding proteins in yeast. PLoS Biol. **2**: e79.

47 Getz, G., E. Levine and E. Domany. 2000. Coupled two-way clustering analysis of gene microarray data. Proc. Natl Acad. Sci. USA **97**: 12079–84.

48 Giraldez, A. J., R. M. Cinalli, M. E. Glasner, et al. 2005. MicroRNAs regulate brain morphogenesis in zebrafish. Science **308**: 833–8.

49 Graber, J. H., C. R. Cantor, S. C. Mohr and T. F. Smith. 1999. *In silico* detection of control signals: mRNA 3′-end-processing sequences in diverse species. Proc. Natl Acad. Sci. USA **96**: 14055–60.

50 Griffiths-Jones, S. 2004. The microRNA Registry. Nucleic Acids Res. **32**: D109–11.

51 Grigull, J., S. Mnaimneh, J. Pootoolal, M. D. Robinson and T. R. Hughes. 2004. Genome-wide analysis of mRNA stability using transcription inhibitors and microarrays reveals posttranscriptional control of ribosome biogenesis factors. Mol. Cell. Biol. **24**: 5534–47.

52 GuhaThakurta, D. and G. D. Stormo. 2001. Identifying target sites for cooperatively binding factors. Bioinformatics **17**: 608–21.

53 Gupta, M. and J. S. Liu. 2005. *De novo cis*-regulatory module elicitation for eukaryotic genomes. Proc. Natl Acad. Sci. USA **102**: 7079–84.

54 Hajarnavis, A., I. Korf and R. Durbin. 2004. A probabilistic model of 3′ end formation in Caenorhabditis elegans. Nucleic Acids Res. **32**: 3392–9.

55 Hallikas, O., K. Palin, N. Sinjushina, R. Rautiainen, E. Ukkonen and J. Taipale. 2006. Genome-wide prediction of mammalian enhancers based on analysis of transcription-factor binding affinity. Cell **124**: 47–59.

56 Handl, J., J. Knowles and D. B. Kell. 2005. Computational cluster validation in post-genomic data analysis. Bioinformatics **21**: 3201–12.

57 Hannenhalli, S. and S. Levy. 2002. Predicting transcription factor synergism. Nucleic Acids Res. **30**: 4278–84.

58 Harbison, C. T., D. B. Gordon, T. I. Lee, et al. 2004. Transcriptional regulatory code of a eukaryotic genome. Nature **431**: 99–104.

59 Hart, C. E., L. Sharenbroich, B. J. Bornstein, D. Trout, B. King, E. Mjolsness and B. J. Wold. 2005. A mathematical and computational framework for quantitative comparison and integration of large-scale gene expression data. Nucleic Acids Res. **33**: 2580–94.

60 Hartemink, A. J., D. K. Gifford, T. S. Jaakkola and R. A. Young. 2002. Combining location and expression data for principled discovery of genetic regulatory network models. Pac. Symp. Biocomput.: 437–49.

61 Hastie, T., R. Tibshirani and J. Friedman. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer, New York, NY.

62 He, L. and G. J. Hannon. 2004. MicroRNAs: small RNAs with a big role in gene regulation. Nat. Rev. Genet. **5**: 522–31.

63 Heinemeyer, T., E. Wingender, I. Reuter, et al. 1998. Databases on transcriptional regulation: TRANSFAC, TRRD and COMPEL. Nucleic Acids Res. **26**: 362–7.

64 Hertz, G. Z., G. W. Hartzell, 3rd and G. D. Stormo. 1990. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. Comput. Appl. Biosci. **6**: 81–92.

65 Hinman, V. F., A. T. Nguyen, R. A. Cameron and E. H. Davidson. 2003. Developmental gene regulatory network architecture across 500 million years of echinoderm evolution. Proc. Natl Acad. Sci. USA **100**: 13356–61.

66 Hobert, O. 2004. Common logic of transcription factor and microRNA action. Trends Biochem. Sci. **29**: 462–8.

67 Holcik, M. and N. Sonenberg. 2005. Translational control in stress and apoptosis. Nat. Rev Mol. Cell. Biol. **6**: 318–27.

68 Holmes, I. and W. J. Bruno. 2001. Evolutionary HMMs: a Bayesian approach to multiple alignment. Bioinformatics **17**: 803–20.

69 Hong, P., X. S. Liu, Q. Zhou, X. Lu, J. S. Liu and W. H. Wong. 2005. A boosting approach for motif modeling using ChIP-chip data. Bioinformatics **21**: 2636–43.

70 Horak, C. E. and M. Snyder. 2002. ChIP-chip: a genomic approach for identifying transcription factor binding sites. Methods Enzymol. **350**: 469–83.

71 Hori, T., Y. Taguchi, S. Uesugi and Y. Kurihara. 2005. The RNA ligands for mouse proline-rich RNA-binding protein (mouse Prrp) contain two consensus sequences in separate loop structure. Nucleic Acids Res. **33**: 190–200.

72 Hughes, J. D., P. W. Estep, S. Tavazoie and G. M. Church. 2000. Computational identification of *cis*-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*. J. Mol. Biol. **296**: 1205–14.

73 Hui, J., L. H. Hung, M. Heiner, S. Schreiner, N. Neumuller, G. Reither, S. A. Haas and A. Bindereif. 2005. Intronic CA-repeat and CA-rich elements: a new class of regulators of mammalian alternative splicing. EMBO J. **24**: 1988–98.

74 Huttenhofer, A., P. Schattner and N. Polacek. 2005. Non-coding RNAs: hope or hype? Trends Genet. **21**: 289–97.

75 Ihmels, J., G. Friedlander, S. Bergmann, O. Sarig, Y. Ziv and N. Barkai. 2002. Revealing modular organization in the yeast transcriptional network. Nat. Genet. **31**: 370–7.

76 Impey, S., S. R. McCorkle, H. Cha-Molstad, et al. 2004. Defining the CREB regulon: a genome-wide analysis of transcription factor regulatory regions. Cell **119**: 1041–54.

77 Istrail, S. and E. H. Davidson. 2005. Logic functions of the genomic *cis*-regulatory code. Proc. Natl Acad. Sci. USA **102**: 4954–9.

78 John, B., A. J. Enright, A. Aravin, T. Tuschl, C. Sander and D. S. Marks. 2004. Human microRNA targets. PLoS Biol. **2**: e363.

79 Johnson, J. M., J. Castle, P. Garrett-Engele, et al. 2003. Genome-wide survey of human alternative pre-mRNA splicing with exon junction microarrays. Science **302**: 2141–4.

80 Jordan, I. K., L. Marino-Ramirez, Y. I. Wolf and E. V. Koonin. 2004. Conservation and coevolution in the scale-free human gene coexpression network. Mol. Biol. Evol **21**: 2058–70.

81 Kaern, M., T. C. Elston, W. J. Blake and J. J. Collins. 2005. Stochasticity in gene expression: from theories to phenotypes. Nat. Rev. Genet. **6**: 451–64.

82 Kampa, D., J. Cheng, P. Kapranov, et al. 2004. Novel RNAs identified from an in-depth analysis of the transcriptome of human chromosomes 21 and 22. Genome Res. **14**: 331–42.

83 Kato, M., N. Hata, N. Banerjee, B. Futcher and M. Q. Zhang. 2004. Identifying combinatorial regulation of transcription factors and binding motifs. Genome Biol. **5**: R56.

84 Kechris, K. J., E. van Zwet, P. J. Bickel and M. B. Eisen. 2004. Detecting DNA regulatory motifs by incorporating positional trends in information content. Genome Biol. **5**: R50.

85 Keles, S., M. van der Laan and M. B. Eisen. 2002. Identification of regulatory elements using a feature selection method. Bioinformatics **18**: 1167–75.

86 Keles, S., M. J. van der Laan and C. Vulpe. 2004. Regulatory motif finding

by logic regression. Bioinformatics **20**: 2799–811.

87 Kirschner, M. W. 2005. The meaning of systems biology. Cell **121**: 503–4.

88 Komar, A. A. and M. Hatzoglou. 2005. Internal ribosome entry sites in cellular mRNAs: mystery of their existence. J. Biol. Chem **280**: 23425–8.

89 Krek, A., D. Grun, M. N. Poy, et al. 2005. Combinatorial microRNA target predictions. Nat. Genet. **37**: 495–500.

90 Ladd, A. N. and T. A. Cooper. 2002. Finding signals that regulate alternative splicing in the post-genomic era. Genome Biol. **3**: REVIEWS0008.1–16.

91 Le, K., K. Mitsouras, M. Roy, Q. Wang, Q. Xu, S. F. Nelson and C. Lee. 2004. Detecting tissue-specific regulation of alternative splicing as a qualitative change in microarray data. Nucleic Acids Res. **32**: e180.

92 Lee, C. and M. Poy. 2004. Analysis of alternative splicing with microarrays: successes and challenges. Genome Biol. **5**: 231.

93 Lee, C. and Q. Wang. 2005. Bioinformatics analysis of alternative splicing. Brief Bioinform **6**: 23–33.

94 Legendre, M. and D. Gautheret. 2003. Sequence determinants in human polyadenylation site selection. BMC Genomics **4**: 7.

95 Levine, M. and E. H. Davidson. 2005. Gene regulatory networks for development. Proc. Natl Acad. Sci. USA **102**: 4936–42.

96 Lewis, B. P., C. B. Burge and D. P. Bartel. 2005. Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. Cell **120**: 15–20.

97 Li, F., T. Long, Y. Lu, Q. Ouyang and C. Tang. 2004. The yeast cell-cycle network is robustly designed. Proc. Natl Acad. Sci. USA **101**: 4781–6.

98 Li, H. R., J. Wang-Rodriguez, T. M. Nair, J. M. Yeakley, Y. S. Kwon, M. Bibikova, C. Zheng, L. Zhou, K. Zhang, T. Downs, X. D. Fu, and J. B. Fan. 2006. Two-dimensional transcriptome profiling: Identification of messenger RNA isoform signatures in prostate cancer from archived paraffin-embedded cancer specimens. Cancer Res. **66**: 4079–88.

99 Lim, L. P., N. C. Lau, P. Garrett-Engele, et al. 2005. Microarray analysis shows that some microRNAs downregulate large numbers of target mRNAs. Nature **433**: 769–73.

100 Liu, C. G., G. A. Calin, B. Meloon, et al. 2004. An oligonucleotide microchip for genome-wide microRNA profiling in human and mouse tissues. Proc. Natl Acad. Sci. USA **101**: 9740–4.

101 Liu, H. X., S. L. Chew, L. Cartegni, M. Q. Zhang and A. R. Krainer. 2000. Exonic splicing enhancer motif recognized by human SC35 under splicing conditions. Mol. Cell. Biol. **20**: 1063–71.

102 Liu, H. X., M. Zhang and A. R. Krainer. 1998. Identification of functional exonic splicing enhancer motifs recognized by individual SR proteins. Genes Dev. **12**: 1998–2012.

103 Liu, X., D. M. Noll, J. D. Lieb and N. D. Clarke. 2005. DIP-chip: rapid and accurate determination of DNA-binding specificity. Genome Res. **15**: 421–7.

104 Liu, X. S., D. L. Brutlag and J. S. Liu. 2002. An algorithm for finding protein-DNA binding sites with applications to chromatin-immunoprecipitation microarray experiments. Nat. Biotechnol. **20**: 835–9.

105 Longabaugh, W. J., E. H. Davidson and H. Bolouri. 2005. Computational representation of developmental genetic regulatory networks. Dev. Biol. **283**: 1–16.

106 Loots, G. G., R. M. Locksley, C. M. Blankespoor, Z. E. Wang, W. Miller, E. M. Rubin and K. A. Frazer. 2000. Identification of a coordinate regulator of interleukins 4, 13, and 5 by cross-species sequence comparisons. Science **288**: 136–40.

107 Mandel-Gutfreund, Y., A. Baron and H. Margalit. 2001. A structure-based approach for prediction of protein binding sites in gene upstream regions. Pac. Symp. Biocomput.: 139–50.

108 Maniatis, T. and R. Reed. 2002. An extensive network of coupling among gene expression machines. Nature **416**: 499–506.

109 McCarroll, S. A., C. T. Murphy, S. Zou, S. D. Pletcher, C. S. Chin, Y. N. Jan, C. Kenyon, C. I. Bargmann and H. Li. 2004. Comparing genomic expression patterns across species identifies shared transcriptional profile in aging. Nat. Genet. **36**: 197–204.

110 Meister, G. and T. Tuschl. 2004. Mechanisms of gene silencing by double-stranded RNA. Nature **431**: 343–9.

111 Meng, Z., P. H. King, L. B. Nabors, N. L. Jackson, C. Y. Chen, P. D. Emanuel and S. W. Blume. 2005. The ELAV RNA-stability factor HuR binds the 5′-untranslated region of the human IGF-IR transcript and differentially represses cap-dependent and IRES-mediated translation. Nucleic Acids Res. **33**: 2962–79.

112 Mignone, F., G. Grillo, F. Licciulli, et al. 2005. UTRdb and UTRsite: a collection of sequences and regulatory motifs of the untranslated regions of eukaryotic mRNAs. Nucleic Acids Res. **33**: D141–6.

113 Moses, A. M., D. Y. Chiang and M. B. Eisen. 2004. Phylogenetic motif detection by expectation-maximization on evolutionary mixtures. Pac. Symp. Biocomput.: 324–35.

114 Mukherjee, S., M. F. Berger, G. Jona, X. S. Wang, D. Muzzey, M. Snyder, R. A. Young and M. L. Bulyk. 2004. Rapid analysis of the DNA-binding specificities of transcription factors with DNA microarrays. Nat. Genet. **36**: 1331–9.

115 Neuwald, A. F., J. S. Liu and C. E. Lawrence. 1995. Gibbs motif sampling: detection of bacterial outer membrane protein repeats. Protein Sci. **4**: 1618–32.

116 O'Donnell, K. A., E. A. Wentzel, K. I. Zeller, C. V. Dang and J. T. Mendell. 2005. c-Myc-regulated microRNAs modulate E2F1 expression. Nature **435**: 839–43.

117 Oliveri, P. and E. H. Davidson. 2004. Gene regulatory network analysis in sea urchin embryos. Methods Cell Biol. **74**: 775–94.

118 Ozbudak, E. M., M. Thattai, H. N. Lim, B. I. Shraiman and A. Van Oudenaarden. 2004. Multistability in the lactose utilization network of *Escherichia coli*. Nature **427**: 737–40.

119 Pan, Q., O. Shai, C. Misquitta, et al. 2004. Revealing global regulatory features of mammalian alternative splicing using a quantitative microarray platform. *Mol. Cell.* **16**: 929–41.

120 Pavesi, G., G. Mauri and G. Pesole. 2004. *In silico* representation and discovery of transcription factor binding sites. Brief Bioinform. **5**: 217–36.

121 Penalva, L. O., S. A. Tenenbaum and J. D. Keene. 2004. Gene expression analysis of messenger RNP complexes. Methods Mol. Biol. **257**: 125–34.

122 Philippakis, A. A., F. S. He and M. L. Bulyk. 2005. Modulefinder: a tool for computational discovery of *cis* regulatory modules. Pac. Symp. Biocomput.: 519–30.

123 Pickering, B. M. and A. E. Willis. 2005. The implications of structured 5′ untranslated regions on translation and disease. Semin. Cell Dev. Biol. **16**: 39–47.

124 Pozzoli, U., L. Riva, G. Menozzi, R. Cagliani, G. P. Comi, N. Bresolin, R. Giorda and M. Sironi. 2004. Over-representation of exonic splicing enhancers in human intronless genes suggests multiple functions in mRNA processing. Biochem. Biophys. Res. Commun. **322**: 470–6.

125 Qin, Z. S., L. A. McCue, W. Thompson, L. Mayerhofer, C. E. Lawrence and J. S. Liu. 2003. Identification of co-regulated genes through Bayesian clustering of predicted regulatory binding sites. Nat. Biotechnol. **21**: 435–9.

126 Qiu, P. 2003. Recent advances in computational promoter analysis in understanding the transcriptional regulatory network. Biochem. Biophys. Res. Commun. **309**: 495–501.

127 Ren, B. and B. D. Dynlacht. 2004. Use of chromatin immunoprecipitation assays in genome-wide location analysis of mammalian transcription factors. Methods Enzymol. **376**: 304–15.

128 Ren, B., F. Robert, J. J. Wyrick, et al. 2000. Genome-wide location and function of DNA binding proteins. Science **290**: 2306–9.

129 Rhodes, D. R., S. Kalyanasundaram, V. Mahavisno, T. R. Barrette, D. Ghosh and A. M. Chinnaiyan. 2005. Mining for regulatory programs in the cancer transcriptome. Nat. Genet. **37**: 579–83.

130 Richter, J. D. and N. Sonenberg. 2005. Regulation of cap-dependent translation by eIF4E inhibitory proteins. Nature **433**: 477–80.

131 Robins, H., Y. Li and R. W. Padgett. 2005. Incorporating structure to predict microRNA targets. Proc. Natl Acad. Sci. USA **102**: 4006–9.

132 Roeder, R. G. 2005. Transcriptional regulation and the role of diverse coactivators in animal cells. FEBS Lett. **579**: 909–15.

133 Sandelin, A., W. Alkema, P. Engstrom, W. W. Wasserman and B. Lenhard. 2004. JASPAR: an open-access database for eukaryotic transcription factor binding profiles. Nucleic Acids Res. **32**: D91–4.

134 Scott, R. E., E. White-Grindley, H. E. Ruley, E. J. Chesler and R. W. Williams. 2005. P2P-R expression is genetically coregulated with components of the translation machinery and with PUM2, a translational repressor that associates with the P2P-R mRNA. J. Cell Physiol. **204**: 99–105.

135 Segal, E., Y. Barash, I. Simon, N. Friedman and D. Koller. 2002. From Promoter Sequence to Expression: A Probabilistic Framework. In Proc. 6th Inter. Conf. on Research in Computational Molecular Biology (RECOMB), Wahsington, DC.

136 Segal, E., M. Shapira, A. Regev, D. Pe'er, D. Botstein, D. Koller and N. Friedman. 2003. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nat. Genet. **34**: 166–76.

137 Sinha, S. 2003. Discriminative motifs. J. Comp. Biol. **10**: 599–615.

138 Sinha, S., M. Blanchette and M. Tompa. 2004. PhyME: a probabilistic algorithm for finding motifs in sets of orthologous sequences. BMC Bioinformatics **5**: 170.

139 Sironi, M., G. Menozzi, L. Riva, R. Cagliani, G. P. Comi, N. Bresolin, R. Giorda and U. Pozzoli. 2004. Silencer elements as possible inhibitors of pseudoexon splicing. Nucleic Acids Res. **32**: 1783–91.

140 Smith, A. D., P. Sumazin, Z. Xuan and M. Q. Zhang. 2006. DNA motifs in human and mouse proximal promoters predict tissue-specific expression. Proc. Natl Acad. Sci. USA **103**: 6275–80.

141 Smith, A. D., P. Sumazin and M. Q. Zhang. 2005. Identifying tissue-selective transcription factor binding sites in vertebrate promoters. Proc. Natl Acad. Sci. USA **102**: 1560–5.

142 Spellman, P. T., G. Sherlock, M. Q. Zhang, et al. 1998. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. Mol. Biol. Cell **9**: 3273–97.

143 Stamm, S., S. Ben-Ari, I. Rafalska, Y. Tang, Z. Zhang, D. Toiber, T. A. Thanaraj and H. Soreq. 2005. Function of alternative splicing. Gene **344**: 1–20.

144 Stamm, S., M. Q. Zhang, T. G. Marr and D. M. Helfman. 1994. A sequence compilation and comparison of exons that are alternatively spliced in neurons. Nucleic Acids Res. **22**: 1515–26.

145 Stamm, S., J. Zhu, K. Nakai, P. Stoilov, O. Stoss and M. Q. Zhang. 2000. An alternative-exon database and its statistical analysis. DNA Cell Biol. **19**: 739–56.

146 Sumazin, P., G. Chen, N. Hata, A. D. Smith, T. Zhang and M. Q. Zhang. 2005. DWE: discriminating word enumerator. Bioinformatics **21**: 31–8.

147 Tabaska, J. E. and M. Q. Zhang. 1999. Detection of polyadenylation signals in human DNA sequences. Gene **231**: 77–86.

148 Tacke, R. and J. L. Manley. 1995. The human splicing factors ASF/SF2 and SC35 possess distinct, functionally significant RNA binding specificities. EMBO J. **14**: 3540–51.

149 Tamada, Y., S. Kim, H. Bannai, S. Imoto, K. Tashiro, S. Kuhara and S. Miyano. 2003. Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. Bioinformatics **19 (Suppl. 2)**: II227–36.

150 Thanaraj, T. A., S. Stamm, F. Clark, J. J. Riethoven, V. Le Texier and J. Muilu. 2004. ASD: the Alternative Splicing Database. Nucleic Acids Res. **32**: D64–9.

151 Thompson, W., M. J. Palumbo, W. W. Wasserman, J. S. Liu and C. E. Lawrence. 2004. Decoding human regulatory circuits. Genome Res. **14**: 1967–74.

152 Tomari, Y. and P. D. Zamore. 2005. Perspective: machines for RNAi. Genes Dev. **19**: 517–29.

153 Tompa, M., N. Li, T. L. Bailey, et al. 2005. Assessing computational tools for the discovery of transcription factor binding sites. Nat. Biotechnol. **23**: 137–44.

154 Van Driessche, N., J. Demsar, E. O. Booth, P. Hill, P. Juvan, B. Zupan, A. Kuspa and G. Shaulsky. 2005. Epistasis analysis with global transcriptional phenotypes. Nat. Genet. **37**: 471–7.

155 Wasserman, W. W., M. Palumbo, W. Thompson, J. W. Fickett and C. E. Lawrence. 2000. Human–mouse genome comparisons to locate regulatory sites. Nat. Genet. **26**: 225–8.

156 Wasserman, W. W. and A. Sandelin. 2004. Applied bioinformatics for the identification of regulatory elements. Nat. Rev. Genet. **5**: 276–87.

157 Wickens, M., D. S. Bernstein, J. Kimble and R. Parker. 2002. A PUF family portrait: 3′UTR regulation as a way of life. Trends Genet. **18**: 150–7.

158 Wilhelm, J. E. and C. A. Smibert. 2005. Mechanisms of translational regulation in Drosophila. Biol. Cell **97**: 235–52.

159 Workman, C. T. and G. D. Stormo. 2000. ANN-Spec: a method for discovering transcription factor binding sites with improved specificity. Pac. Symp. Biocomput.: 467–78.

160 Wray, G. A., M. W. Hahn, E. Abouheif, J. P. Balhoff, M. Pizer, M. V. Rockman and L. A. Romano. 2003. The evolution of transcriptional regulation in eukaryotes. Mol. Biol. Evol. **20**: 1377–419.

161 Wu, Q., T. Zhang, J. F. Cheng, et al. 2001. Comparative DNA sequence analysis of mouse and human protocadherin gene clusters. Genome Res. **11**: 389–404.

162 Xie, X., J. Lu, E. J. Kulbokas, T. R. Golub, V. Mootha, K. Lindblad-Toh, E. S. Lander and M. Kellis. 2005. Systematic discovery of regulatory motifs in human promoters and 3′ UTRs by comparison of several mammals. Nature **434**: 338–45.

163 Yeakley, J. M., J. B. Fan, D. Doucet, L. Luo, E. Wickham, Z. Ye, M. S. Chee and X. D. Fu. 2002. Profiling alternative splicing on fiber-optic arrays. Nat. Biotechnol. **20**: 353–8.

164 Zhang, C., H. R. Li, J. B. Fan, J. Wang-Rodriguez, T. Downs, X. D. Fu and M. Q. Zhang. 2006. Profiling alternatively spliced mRNA isoforms for prostate cancer classification. BMC Bioinformatics **7**: 202.

165 Zhang, M. Q. 2002. *Computational Methods for Promoter Recognition*. MIT Press, Cambridge, MA.

166 Zhang, M. Q. 2002. Computational prediction of eukaryotic protein-coding genes. Nat. Rev. Genet. **3**: 698–709.

167 Zhang, M. Q. 2003. Prediction, annotation, and analysis of human promoters. Cold Spring Harb. Symp. Quant Biol. **68**: 217–25.

168 Zhang, X. H. and L. A. Chasin. 2004. Computational definition of sequence motifs governing constitutive exon splicing. Genes Dev. **18**: 1241–50.

169 Zhao, F., Z. Xuan, L. Liu and M. Q. Zhang. 2005. TRED: a Transcriptional Regulatory Element Database and a platform for *in silico* gene regulation studies. Nucleic Acids Res. **33**: D103–7.

170 Zheng, Z. M. 2004. Regulation of alternative RNA splicing by exon definition and exon sequences in viral and mammalian gene expression. J. Biomed. Sci. **11**: 278–94.

171 ZHU, J. AND M. Q. ZHANG. 1999. SCPD: a promoter database of the yeast Saccharomyces cerevisiae. Bioinformatics **15**: 607–11.

172 ZHU, Z., Y. PILPEL AND G. M. CHURCH. 2002. Computational identification of transcription factor binding sites via a transcription-factor-centric clustering (TFCC) algorithm. J. Mol. Biol. **318**: 71–81.

22
Modeling Cell Signaling Networks

Anthony Hasseldine, Azi Lipshtat, Ravi Iyengar and Avi Ma'ayan

1 Introduction

A cell's signaling network is in many ways analogous to an animal's nervous system, connecting and directing the activities of its constituent parts, and processing information from the outside world; and, as in neurobiology, despite extensive study, there is still a lot to learn about cell signaling. The reason for the intricacy of signaling networks is evident if one considers that they are responsible for the control of cellular functions and information processing, and that such control systems are often more complex than the core functions they regulate. To borrow a classical example from Kuipers [33] and Lauffenburger [36], consider a simple system in which water delivery is required from a tank with in- and outflow pipes. This is a primitive core function indeed, but to prevent accidental overflow or depletion, it still requires a control system comprised of valves, flow meters, transducers and regulators, used in combination according to an appropriate computational algorithm, i.e. a relatively complex control system. Cells, which are considerably more sophisticated than water tanks, have vastly more complex control systems (signaling networks), presenting a considerable challenge in developing detailed predictive models. Different problems call for different approaches. Hence, in this chapter we outline a few complementary modeling strategies and the circumstances in which they may be useful.

1.1 Components and Cascades

The signal transduction field was born with the discovery of adenosine 3′,5′-cyclic monophosphate (cAMP) by Earl Sutherland and coworkers in the 1950s [58]. They called this molecule a "second messenger" in order to differentiate it from extracellular "first messengers" (hormones and neurotransmitters), which were the only real signaling molecules known at that time. Subsequently, other components of the cAMP signaling pathway were identified. In particular, the discovery that hormones, such as epinephrine and glucagon,

Bioinformatics – From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

do not act by binding directly to adenylylcyclases. It was later discovered that receptors and GTP-binding proteins (G-proteins) are required to mediate such response. This finding led to the formulation of the "signal transduction" concept, inspired partly by the burgeoning field of cybernetics [52].

In addition to cAMP, several other small molecules also play important roles in signal transduction. From other work in the 1950s, it was noted that membrane phospholipid metabolism was altered in response to acetylcholine treatment of cells [23]. This response involves inositol phospholipids and consequently the products of their hydrolysis, i.e. phosphoinositols (e.g. IP_3, IP_4) and diacylglycerol (DAG), are among the best characterized lipid-derived signaling molecules. Arachidonic acid, another lipid metabolite, is also highly important in a variety of signaling events, both intra- and intercellularly.

Some small molecules signal via the production or release of other small molecules. For example, adenylylcyclase, an enzyme that catalyzes the production of the small messenger cAMP, causes an indirect activation of protein kinase A (PKA). cAMP binds to the regulatory subunit of PKA and this binding reaction causes the release of the catalytic subunit from the regulatory subunit. The catalytic subunit of PKA is then free to diffuse, and can now catalyze the phosphorylation, potentiation and activation of ion Ca^{2+} channels. Ca^{2+}, via its binding protein, calmodulin (CaM), can also activate, for example, certain isoforms of phosphodiesterases and the phosphatase, calcineurin. However, the most common effect of small-molecule release or synthesis is the regulation of protein kinase activity. The best known of these are cAMP-dependent protein kinase (PKA), cGMP-dependent protein kinase (PKG), the DAG-sensitive kinases PKC and PKD, the inositol phospholipid-sensitive PKB/Akt, and the calcium-CaM-dependent kinases (CaMK I–IV).

Protein phosphorylation is a very common regulatory modification in cell signaling. Most protein kinases listed above are themselves regulated by phosphorylation, while protein kinases of another important family, the mitogen-activated protein kinases (MAPKs), are activated by phosphorylation alone, in the classic MAPKKK–MAPKK–MAPK cascade.

In addition to small-molecule sensitivity and phosphorylation, there are also many regulatory protein–protein binding interactions in cell signaling. Many of these involve another important category of molecule: G-proteins. G-proteins fall into two categories: heterotrimeric G-proteins (e.g. G_q, G_s, G_i and G_z), which are activated by cell surface receptors of the seven-transmembrane (7TM) receptor superfamily, and monomeric G-proteins ("small G-proteins", e.g. Ras, Rho, Rab), which are activated by specific guanine nucleotide exchange factors (GEFs). In both cases, the GTP-bound form is the active state and G-protein activation is terminated by its hydrolysis to GDP. For small G-proteins, this is usually catalyzed by a GTPase-activating-protein (GAP), while the intrinsic GTPase rates of heterotrimeric G-proteins may be boosted

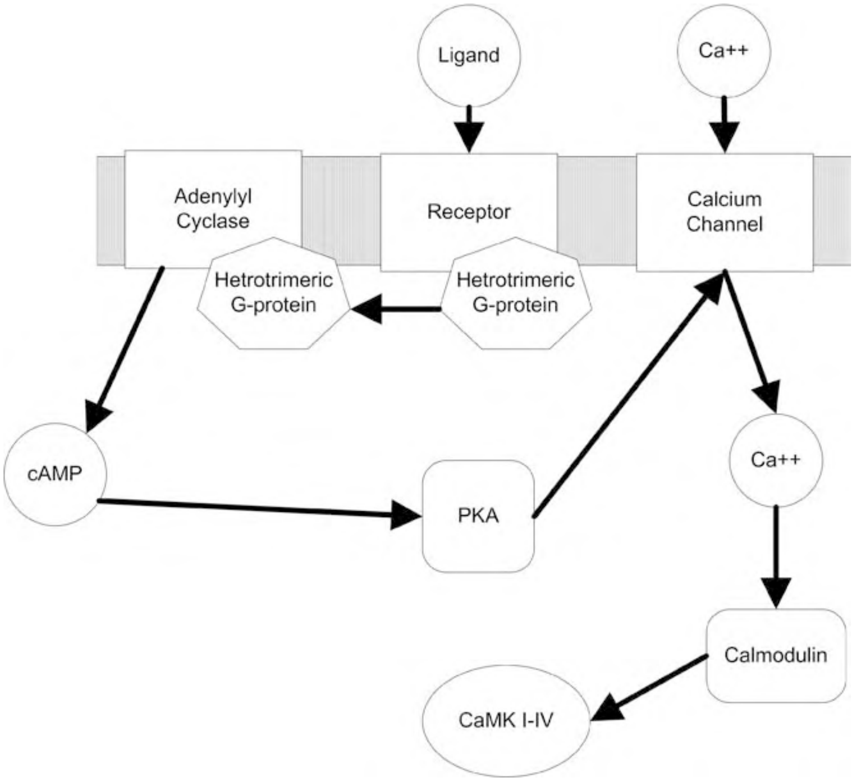


Figure 1 Canonical signaling pathway to illustrate how increase in production of one small-molecule second messenger (cAMP) can lead to an increase in intracellular concentration of another second messenger (Ca^{2+}). Here, an example from heart cells is shown. The signal starts by the binding of some extracellular ligand to a transmembrane cell surface GPCR. The binding of the ligand induces a conformational change in the receptor structure, which results in the activation of a heterotrimeric G-protein. The heterotrimeric G-protein activates the membrane-bound enzyme adenylylcyclase. This enzyme catalyzes the formation of the second messenger cAMP, which in turn diffuses through the cytoplasm to bind the regulatory subunit of protein kinase PKA. This binding of cAMP to the regulatory subunit of PKA results in the release of the catalytic subunit, which now is able to diffuse and phosphorylate Ca^{2+} channels, and potentiate their open state to induce intracellular Ca^{2+} entry. Intracellular Ca^{2+} can bind to a calcium sensor protein named CaM which can bind to and activate CaMK-type family of CaM-activated kinases.

by the GAP activity of RGS proteins (regulators of G-protein signaling). See Figure 1 for a canonical cascade of signaling interactions using some of the components and interactions described.

The conformational change that results from a binding interaction often causes a change in the target’s catalytic activity. 7TM receptor activation by an agonist, permitting the receptor to catalyze G-protein activation, is an example of this. Alternatively, the conformational change may result in the concealment or exposure of other binding sites on the target, thus regulating protein complex formation, for example. Autoinhibitory domains and subunits are common; PKA, for instance, is activated by the binding of cAMP to its regulatory subunits, causing them to dissociate from the catalytic subunits, which may then phosphorylate its targets.

Most of the protein kinases described above target serine or threonine residues for phosphorylation, but tyrosine kinases are also very important in the signal transduction systems of higher organisms, particularly in the regulation of growth [44]. They come in two classes: receptor tyrosine kinases, which are transmembrane proteins sensitive to growth factors, neurotrophins and/or insulin, and nonreceptor tyrosine kinases, such as Src, Jak and FAK.

Counterbalancing the effects of protein kinases are protein phosphatases. Serine/threonine phosphatases are far fewer in number than serine/threonine kinases and appear to be regulated less extensively. Nonetheless, some, such as the calcium-CaM-dependent calcineurin, do regulate the activity of phosphorylated substrates in a signal-dependent manner. Tyrosine phosphatases are well known to be important regulators of cell signaling – many, such as the T cell marker CD45, are in fact cell surface receptors.

There are also some protein kinases and phosphatases that are known as "dual specificity", because they catalyze the addition or removal of phosphate groups from both serine/threonine and tyrosine residues.

The molecules detailed above comprise the basic building blocks of the canonical signal transduction network: several types of small molecule (e.g. cAMP, Ca^{2+} and the phosphoinositol IP_3), G-protein-coupled receptors (GPCRs/7TM receptors), heterotrimeric G-proteins, receptor tyrosine kinases, nonreceptor serine/threonine and tyrosine kinases, dual-specificity kinases, serine/threonine, tyrosine and dual-specificity phosphatases, lipid kinases (although not mentioned explicitly, these produce phospholipids), phospholipases (which hydrolyze phospholipids), small G-proteins, GEFs, GAPs, and nucleotide cyclases. More generally, the kinetic building blocks of cell signaling are small-molecule–protein and protein–protein binding interactions, and enzymatic reactions.

1.2 From Pathways to Networks

1.2.1 Interactions between Signaling Pathways

With the discovery of each of the signaling components described above, much effort was focused on discovering a corresponding functional role. This process was, to a large extent, informed by the original view of signal transduction as a linear cascade for information transfer and signal amplification. However, even from the earliest biochemical and cell biological research,

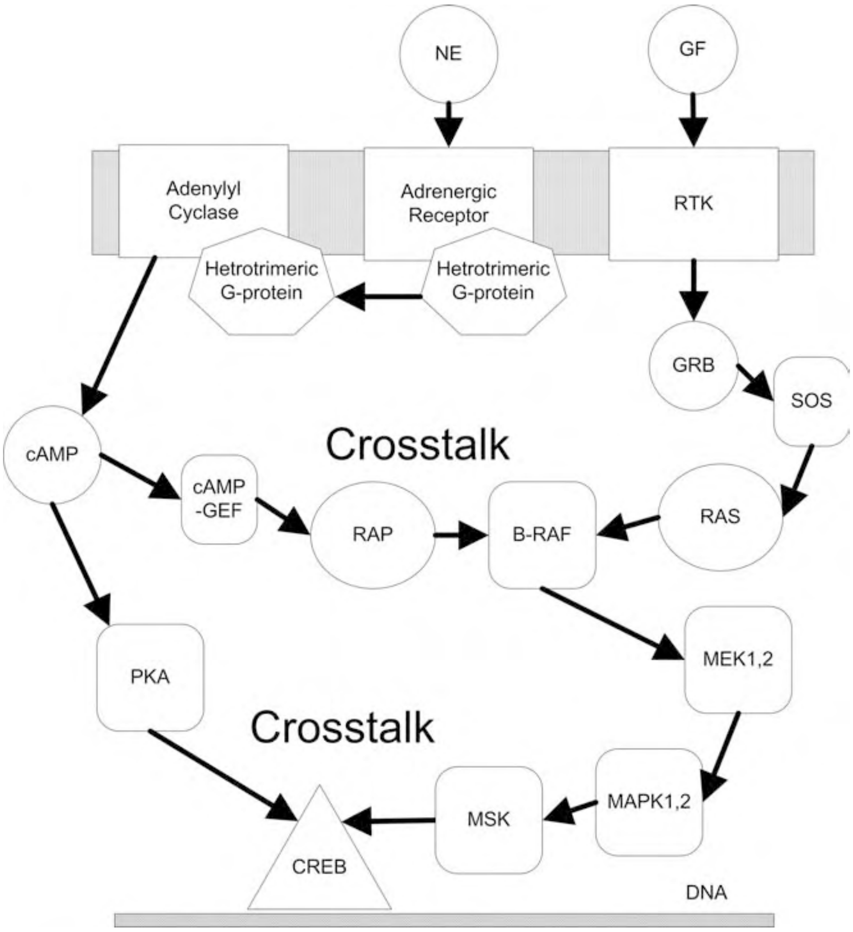


Figure 2 Illustration of "cross-talk" between two canonical signaling pathways: the cAMP–PKA pathway and the MAPK pathway. Here, an example from neurons is shown. Norepinephrine (NE) is a ligand that can bind to adrenergic GPCR receptor types. This induces the cAMP cascade (see for Figure 1). cAMP is also known to activate an exchange factor (cAMP-GEF) that can influence the MAPK cascade through the activation of the small G-protein Rap. The MAPK cascade is typically induced by the binding of growth factors (GF) to cell surface family of receptors named receptor tyrosine kinases (RTK). Once ligand bound, these receptors dimerize, autophosphorylate each other, and can bind and phosphorylate intracellular proteins such as GRB. GRB once bound to the receptor can bind to SOS which binds and activates the small G-protein RAS. RAS in turn activates MAPKKK (B-RAF) which is in turn phosphorylates and activates MAPKK (MEK1,2) which phosphorylates and activates MAPK1,2. Both MAPK activated kinase MSK and PKA can phosphorylate and activate the transcription factor CREB. CREB forms homodimers or heterodimers (with other CREB family members) when it is in its phosphorylated form. Active CREB translocates into the nucleus, binds to DNA and affects gene transcriptional regulation.

it was clear that pathways did not really operate independently, and the interactions between them were termed "cross-talk".

Examples of interactions between signaling pathways can be found at many levels, from receptors to effectors. For example, the two endogenous adrenergic agonists, epinephrine and norepinephrine, activate one or several (depending on the cell type) of 10 different α- and β-adrenergic receptors, which differ in their regulation and also, by class (α_1, α_2, β), in their G-protein/effector coupling. Other hormone and neurotransmitter receptors may affect adrenergic receptor sensitivity (e.g. by heterologous desensitization), and also share some of the same effector pathways, which in turn are further connected at multiple levels downstream. See Figure 2 for two examples of "cross-talk" between two canonical signaling pathways – the cAMP–PKA pathway and MAPK pathway.

Interactions between pathways result in regulatory signaling computation, which is limited in "linear" signal cascades. Much as signaling pathways themselves connect and control core cellular functions, these interactions between pathways, while not central to the task of getting information through the cell, are no doubt critical for sophisticated control of cellular behavior. Note, however, that this distinction between pathways and their interactions is contextual, because many signaling components are multifunctional. This point is further expanded below.

1.2.2 Implications of Network Topology

The growth in understanding signaling pathways and interconnections, and the advent of high-throughput techniques for studying large-scale protein interactions and protein post-translational modifications (i.e. phosphorylation) have led to the mapping of intracellular interactions on a breadth never considered before. One striking characteristic of the networks that have emerged from assembling many interactions to form *in silico* networks data sets is the so-called "small-world" topology; this type of organization means that on average there are a small number of connection "steps" between any two components (nodes), as well as high clustering (the proportion of a node's interacting partners that interact with one another) compared with the same statistics computed for random or shuffled networks [64]. This has interesting biological implications. Short paths across the signaling network would presumably be advantageous for rapid signal transfer, while high clustering implies the existence of network motifs, which can confer the ability to process information as it is being transferred.

The second striking topological property that emerges in complex cell signaling networks is their nodal connectivity degree distribution which is referred to as being "scale-free". The distribution of nodes with the same number of neighbors (number of direct biochemical interacting partners) commonly follows a power law [4]. That is, there are many nodes with very few connections and fewer, but substantial, numbers of nodes with many connections (the distribution has a long heavy tail). In terms of the components and cascades described above, it is easy to see that there are several candidate proteins for "hubs", i.e. highly connected nodes such as CaM, PKA and PKC. Graph theory tells us that, while the eradication of most nodes is tolerated easily, hubs are very important for maintaining network structure. This conclusion might appear trivial in light of the examples above – one does not need graph theory to conclude that PKA or PKC are important signaling molecules since they are known to be involved in almost all cell signaling processes and have many phosphorylation targets. Many other hubs are not so well characterized, however, and understanding network vulnerability has clear implications for understanding and identifying drug targets and treating disease. The appearance of power law-distributed networks in many complex systems abstracted to networks, including biology, has been proposed to have come about by "highly optimized tolerance" [10]. This term was coined for a design or evolutionary process that confers robustness to a wide variety of anticipated or experienced conditions. The trade-off is that these networks are very vulnerable to unforeseen or novel perturbations or attacks; again, the pathophysiological implications of this may be far reaching for drug discovery.

Network/graph theory is in itself a way of modeling cell signaling pathways [9], but it is a technique that is still in its infancy and one that can yield only a qualitative picture. It should also be noted that questions have been raised about the data used to draw some of the above conclusions, in particular, the validity of the interactions from yeast-two-hybrid and proteomics studies [30, 63] (see also Chapters 28 and 31). However, similar network characteristics have also been observed in data sets painstakingly collected from older studies in the scientific literature that used less error-prone techniques [40]. Of course, manually collected data also have their limitations, being the product of research driven by human interest and hypotheses, but the similarity in topological properties that emerges from the two approaches is encouraging.

1.2.3 Network Motifs

A group of connections or direct interactions between few components in the cell signaling network that has the potential to perform an information-processing function may be described as a regulatory network motif. Probably the best known is the negative feedback loop (NFBL), which occurs often in metabolic pathways and in desensitization, as in the case of G_s-coupled receptor phosphorylation by PKA and GRK (G-protein-coupled receptor kinase) (Figure 3). The former forms a five-node NFBL which permits the

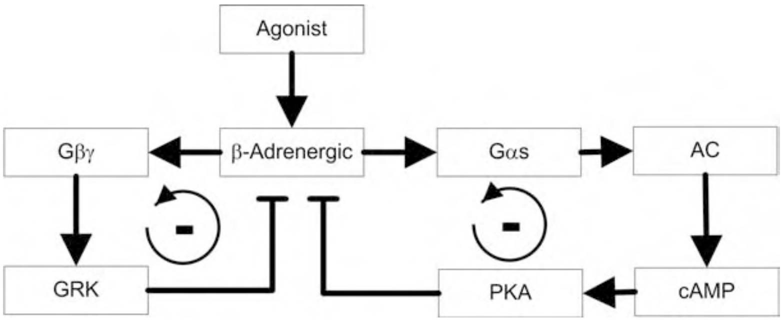


Figure 3 The NFBLs that regulate the sensitivity of a G_s-coupled receptor. Stimulatory connections are symbolized by arrows and inhibitory connections by plungers. Agonist/ligand binds to the receptor which activates G_s, causing it to dissociate to $G\alpha_s$ and Gβγ subunits. GRK is recruited to the plasma membrane by Gβγ subunits, where it phosphorylates and uncouples the active receptor. PKA, which is activated by increased cAMP concentrations, also phosphorylates the receptor. This may be caused by activation of this or another G_s-coupled receptor, or by other stimuli that activate adenylylcyclase (AC).

cell to adjust its sensitivity to the extracellular environment according to the concentration of a second messenger (cAMP) that is downstream of the receptor. This allows for modulation of receptor sensitivity by other detector systems that are coupled to cAMP. GRK, by contrast, is recruited directly by the heterotrimeric G-protein Gβγ and regulates receptor sensitivity in direct proportion to receptor activation. Hence, these two mechanisms, in concert, confer the ability to tightly regulate receptor sensitivity according to the cell's experience. In general terms, negative feedback is a mechanism for setting cellular expectations and preventing signaling hyperactivity.

In contrast to NFBLs, positive feedback loops (PFBLs) can produce *bistability* (or *switching*). Bistability, as the name suggests, is the ability of a system to stably occupy one of two steady states, with very little chance of coming to rest at an intermediate activity level. This can occur if there is an even number of inhibitory interactions, producing "double-negative" regulation or no negative interactions (all the links in the feedback loop are positive). There are several examples of human engineered systems which this motif has been engineered [16].

Switching behavior and multistability are often associated with hysteresis [37]. PFBLs are known to be involved in amplification of small stimuli and the triggering of cellular state changes. A similar biochemical switch has also been proposed: "ultrasensitivity" emerges readily from a signaling cascade, such as the sequential phosphorylation of kinases leading to MAPK activation [7, 8]. Like the bistability PFBL motifs described above, this is a threshold mechanism, but in the case of an ultrasensitive cascade, the switching need only be in one direction such that the PFBL is internal to one enzyme. Of course, ultrasensitive cascades can exist in combination with other motifs, which may confer bidirectional switching overall.

The network motifs described above have long been known, at least in terms of regulatory negative feedback inhibition, and amplification by ultrasensitivity and positive feedback created by a group of coupled biochemical reactions, but the concept of switching is relatively new in cell signaling. Experimental methods such as immunoblotting to observe state changes often do not produce evidence for this behavior, because switching occurs at the single-cell level which is below the resolution of most experimental techniques, i.e. bistability in cell populations, be they in a tissue or a culture dish, will be hidden by averaging. The result of a stimulus is an apparently gradual change in behavior, reflecting a cumulative distribution of single-cell threshold responses.

One important contribution of cellular switches is that they comprise a means of producing cellular state by consolidating many inputs into a single output response. This might be compared with decision making, classification and information processing. Hence, the ubiquitous MAPK cascade (Figure 2), which controls a broad range of cellular outcomes, e.g. by regulating the machinery required for differentiation or proliferation, is not only ultrasensitive, but displays bistability because of a PFBL between MAPK activation and arachidonic acid-mediated PKC activation [7,8].

An additional implication of bistability and ultrasensitivity is a reduction in the number of possible steady states. The signaling network is constrained to a given range of presumably dynamical stable steady-state configurations, improving response reproducibility and network stability to perturbations. This analog-to-digital conversion is obviously not desirable in all regions of the cellular regulatory network, but may be particularly advantageous in the regulation of key effector systems. The term "attractor" is used by mathematicians to describe one of these small volumes in "state space" that biochemical regulatory networks tend to occupy [25].

In addition to the well-studied PFBL and NFBL network motifs outlined above, two other motifs, feedforward loops (FFLs) and *bifans*, have recently been found to be statistically enriched in cell signaling regulatory networks compared with network motifs identified in randomized or shuffled networks. These additional two motifs are potentially important for information processing capabilities in biological networks (these motifs are diagrammed in Figure 4). The FFL has been studied most intensively in gene regulatory networks (GRNs), where it has been proposed to perform several biologically interesting functions that can be extrapolated to cell signaling. FFLs have been categorized into two subdivisions: *coherent* and *incoherent* [41]. The former refers to FFLs in which regulation is consistent across the two arms of the motif, i.e. the node furthest downstream (output node) receives the same

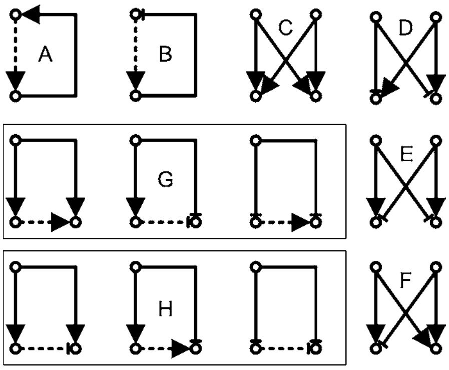


Figure 4 Network motifs. Nodes are symbolized by circles, stimulatory connections by arrows and inhibitory connections by plungers. A dashed line indicates that the connection is not necessarily direct. (A) PFBL. (B) NFBL. (C) "All-positive" bifan. (D) Bifan with "accelerator" and "brake" arrangement. (E) "Competitive" bi-fan. (F) A "biased" bifan (three negative and one positive is another possible configuration). (G) Examples of coherent FFLs (i.e. consistent influence on the output across both arms). (H) Examples of incoherent FFLs (in which conflicting signals are sent across the two arms).

type of signal from both its input arms (Figure 4). In contrast, incoherent FFLs are those in which the two arms have antagonistic effects on the activity of the downstream node. A FFL can modulate the duration of signaling, conferring either signal-onset delay or signal-decay acceleration, depending on its configuration. Detail analysis of different FFL configurations and their potential dynamical behavior was developed by Alon and coworkers [14, 41–43].

The bifan motif, like the FFL, plays a potentially important role in signal integration, although it has not yet been studied extensively. In principle, there are several functions it could perform (see Figure 4 for diagrams). "All-positive links" or "all-negative links" bifans would be expected to act as coincidence detectors and branching points, to produce consistent cellular responses from multiple inputs. The all-negative and all-positive bifan configuration is highly abundant in cell signaling networks most likely because cellular components (proteins) arise in these networks through duplication–divergence evolutionary processes. Regulation in tandem may also be achieved when upstream nodes are connected as follows: positive–positive ("accelerator" input) and negative–negative ("brake" input), with each output receiving one of each. This configuration is probably the most common, biologically, due to the tendency for kinases and phosphatases, as well as GEFs and GAPs, to share substrates. Alternatively, if the inputs each have one positive and one negative connection to the output nodes, the motif would be expected to function as two competing cascades. This could occur under circumstances in which phosphorylation stimulated one pathway and inhibited the other. There is also the "biased" bifan, with just one connection differing in sign from the other three; this is expected to produce coincidence detection at one output and competition at the other. The information-processing function of each of these motifs will further depend on the nature of the inputs' interactions, be they competitive, conditional or synergistic. In Boolean algebra or in electrical engineering parlance, it depends whether the inputs form OR or AND gates.

Other motifs, and more analysis of their functions, have been reported, particularly by Arkin and coworkers. These include signaling configurations that act as oscillators, various frequency filters and amplitude filters [3, 56, 65]. With further research, it will become clearer which of these are utilized in nature. Such analyses may provide some interesting insights into evolution's engineering principles of biological systems.

2 Types of Models and the Information they can Yield

2.1 Boolean Networks and Bayesian Networks Modeling Approaches

The broadest analysis that can be applied to a signaling network is the top-down approach employed by graph (network) theory. This approach yields a description of the entire network in terms of its topology, as well as measures such as average number of connections per component and maximum distance across the signaling network (see Section 1.2.2). This is a rapidly growing new field, from which many novel applications and techniques are exponentially emerging. For example, the analysis of "dynamic networks" may deliver biologically relevant insights into the way in which signaling and GRNs are rewired in response to external cues and the cell cycle [5, 39]. Furthermore, techniques from the field of machine learning have recently been employed in an attempt to derive the evolutionary foundation of biological networks by analyzing the frequency of the motifs they contain [47]. None of these techniques, however, yet offer a dynamic simulation system for predicting and investigating the function of cell signaling network interactions.

Boolean networks modeling is the simplest dynamical depiction of a signaling system, representing states of components as "1" and "0" [31]. The activation state of a node is governed by that of its input neighbors using Boolean logic (Boolean function). Activation and inhibition in a linear cascade can be represented by activation links and inhibition links, while there are eight possible Boolean functions to decode two Boolean inputs, OR and AND suffice to be used to simulate the effects of two converging inputs in signaling networks. For diagrams, formulas and an explanation of these, see Figure 5.

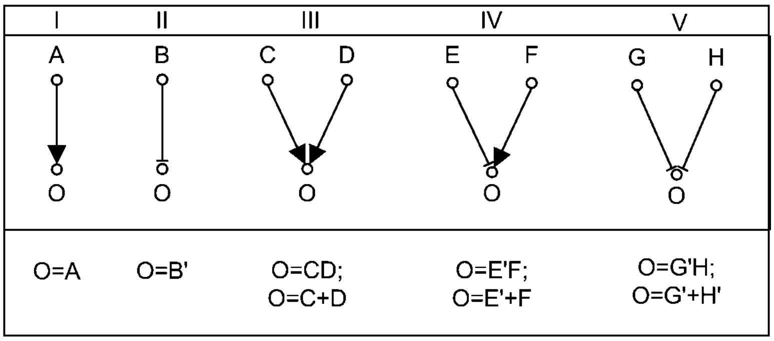


Figure 5 Boolean representations of signaling interactions. Components (nodes) are represented by circles, each labeled with a letter: A–H are inputs and O is the output. Stimulatory connections are symbolized by arrowheads and inhibitory connections by plungers. The two unary functions (single input), I and II, are IF (O = A) and NOT (O = B′), respectively; in Boolean notation, the prime symbol represents the *complement* (it "flips" a value between "0" and "1"). The other three functions are two-input Boolean functions. In III, two stimulatory inputs converge on an output; this interaction may be represented by an AND function (O = CD) if both inputs are required to activate the output and by an OR function (O = C + D) if only one input is required to propagate the signal. In IV, dual regulation (positive and negative) of a node is shown. The AND function (O = E′F) describes this interaction if the inhibitor is off and the activator is on. Whereas the OR function (O = E′ + F) describes activation either when the inhibitor is off OR the activator is on. V shows a binary interaction in which two inhibitors converge on the output. Again, AND (O = G′H′) and OR (O = G′ + H′) functions can be used to represent the interaction; in the former case, both inputs need to be inactive, for the output to be turned on. In the latter case, both inputs need to be activated to inhibit the output.

The Boolean approach is useful for handling very large models, as it is computationally cheap compared with other dynamical modeling approaches such as ordinary differential equations (ODEs) and partial differential equations (PDEs) representations. It is particularly useful when the relationships between nodes have not been characterized quantitatively, i.e. in systems with unknown kinetics of binding and catalysis. Pseudodynamics modeling can also be used to create dynamic "maps" of signaling network topology, which reveal the presence of information-processing motifs specific to particular pathways or regions of signaling space [40]. While quick and easy, relative to other modeling strategies, Boolean and pseudodynamics representations of cell signaling do have limitations. Imposition of binary descriptions can at times be somewhat arbitrary, as many complex, conditional activation events exist in cell signaling. Some of these issues are being addressed by ongoing work in the field. The introduction of more complex functions, customized for every node based on careful experimental observation [11], may go some way towards ameliorating these problems.

Cell signaling networks could also be modeled by classifier systems [24]. Statistical classification machine learning-based models are called Bayesian networks. Bayesian networks are most commonly used to build GRNs from microarray time series or single knockout data sets [18, 53, 66]. Bayesian networks are acyclic graphs where nodes represent variables and links are probabilistic influences of variables on each other. Bayesian networks can be useful in interpreting multidimensional signalome type experimental data such as phospho-proteomics. For example, a Bayesian network was built to understand mouse embryonic stem cells cellular choices under different stimulations protocols [51, 65]. These cells can be driven to self-renewal or to differentiation in culture based on extracellular media provided. A similar approach was used to study the relationships between proteins and phospholipids, and the directionality of their links, after T cell activation of naive T cells [54]. Using data from single-cell measurements using flow cytometry to measure the phosphorylation levels of key signaling components [26] it is possible to determine the hierarchical ordering of signaling components by applying experimental perturbations, such as knocking out genes and proteins, by either pharmacological agents or RNA interference. Another relatively simple qualitative related statistical method was used to study cell signaling axes of apoptosis [28].

2.2 Quantitative Dynamics Modeling

In order to understand intracellular behavior on a quantitative detailed level, quantitative system dynamics need to be captured. Although certain behaviors are fairly robust with respect to kinetic parameters, there are also frequent examples of systems-level behaviors that emerge from quantitative dynamics rather than simply connection maps. An example of this is a system alluded to earlier, in which bistability results from a PFBL formed by the MAPK cascade in conjunction with arachidonic acid production and PKC activation (see Figure 6). This switch is in turn regulated by a PFBL between MAPK and MAPK phosphatase (MKP): active MAPK phosphorylates MKP, protecting it from degradation, while MKP activity inhibits MAPK. Once enough MKP has accumulated, its inhibitory influence attenuates the activity of the PFBL and the MAPK response becomes proportional to the input stimulus, rather than sustained or switch-like (Figure 6).

Kinetics-dependent behavior is also evidenced in the activation of immediate early genes by MAPK – an example of a coherent FFL. MAPK can stimulate transcription and translation of c-Fos, and it can also phosphorylate the c-Fos protein, which protects it from degradation by the ubiquitin–proteasome system. However, the duration of MAPK activation is critical for the function of the motif. If it is brief, then protection and synthesis are asynchronous,

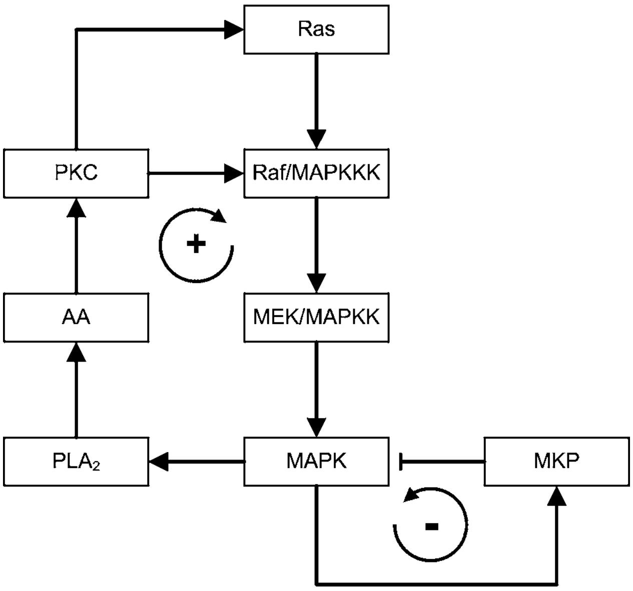


Figure 6 MAPK regulatory modules include both PFBLs and NFBLs. The former includes the small G-protein, Ras, which activates the MAPKKK, Raf, which in turn activates the MAPKK, MEK1,2, which activates MAPK1,2. MAPK1,2 stimulates phospholipase A_2 (PLA_2), which synthesizes arachidonic acid (AA), activating PKC, which in turn can promote both Ras and Raf activation. This PFBL confers bistability of the system in the absence of high levels of MKP. The NFBL involves increased synthesis of MKP and its protection from degradation, both of which are stimulated by MAPK. Dephosphorylation inhibits MAPK, so high levels of MKP, which can come about either through persistent activation of MAPK as outlined above, or via other regulators, prevent the sustained activation of MAPK despite the positive feedback loop, thus eliminating the system's bistability.

meaning that the FFL is ineffective, i.e. if MAPK activity has returned to a low level in the time required for transcription and translation, the new c-Fos is not phosphorylated and is consequently degraded. By contrast, when MAPK activation is sustained, the FFL is functional, causing an increase in c-Fos activity (Figure 7) [15, 50].

Signal processing can also be affected simply by an activation cascade that works across divergent time scales. For example, the dynamics of calcium concentration changes (due to channel fluxes and diffusion) is faster than that of calcium detection systems (small-molecule–protein binding interactions), which in turn operate faster than the downstream protein–protein interactions and enzymatic processes that rely upon calcium detection. This change in dynamics can act as a frequency filter, typically low pass, which removes noise from the external input signal. Dynamic quantitative models can capture these traits, whereas Boolean representations probably cannot.

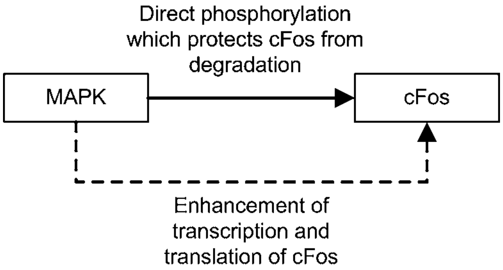

Direct phosphorylation which protects cFos from degradation
MAPK
cFos
Enhancement of transcription and translation of cFos

Figure 7 Example of a PFFL where MAPK directly phosphorylates c-Fos and protects it from degradation, while at the same time it can enhance c-Fos transcription and translation through indirect pathways. Sustained activation of MAPK is required to achieve enhanced c-Fos activity.

2.2.1 **Deterministic Models**

Signaling reactions are discrete events, involving binding and/or chemical transformation of individual molecules, sometimes catalyzed by other individual molecules. Predicting the exact details at a molecular level requires calculations of individual probability trajectories, and is currently not feasible for systems with more than just a few molecules. However, if there are many reacting molecules, their average behavior can be described by some relatively simple mathematics, in which concentrations are treated as continuous variables.

The reactions in a signaling network can be broken into two categories: binding and/or catalyzed transformation; in most cases, the latter process is irreversible or may be considered irreversible for the purposes of modeling. In cases where some details are not known, approximations can suffice. The most commonly used such approximation is the Michaelis–Menten formulation [46], that assumes that the mass-action is significantly faster than the catalyzed transformation, and so the forward and reverse reaction rate constants, k_f and k_r, collapse into one term, K_M; enzyme activity is also described by the constants V_max, which is the product of total enzyme concentration and k_cat. The use of V_max in the Michaelis–Menten equation can be particularly advantageous to the modeler, as this parameter is more easily measured and is found often in the biochemistry and molecular biology literature compared with enzyme concentrations and k_cat. See Box 1 for details of these reactions and the expression of their kinetics. Other customized kinetic formalisms are also possible, so long as the reaction can be expressed by an ODE. Chapter 20 also discusses the kinetics of biochemical reactions.

Box 1 Basic reactions and the deterministic expression of their kinetics

$$A + E \leftrightarrow AE \tag{1}$$

$$AE \rightarrow B + E \tag{2}$$

The rate equations (ODEs) that describe these reactions deterministically are, respectively:

$$\frac{\mathrm{d}[AE]}{\mathrm{d}t} = [A][E]k_{\mathrm{f}} - [AE]k_{\mathrm{r}} - [AE]k_{\mathrm{cat}} \tag{3}$$

$$\frac{\mathrm{d}[B]}{\mathrm{d}t} = [AE]k_{\mathrm{cat}} \ , \tag{4}$$

where k_{f} and k_{r} are the forward and reverse rate constants, respectively, having units of concentration^{-1} time^{-1} and time^{-1}, respectively (herein, following convention, we use µM and s); k_{cat} is the first order rate constant of catalysis (s^{-1}); thus all reaction rates are expressed in units of µM s^{-1}.

Combining Eqs. (3) and (4), and making the assumption that the concentration of enzyme–substrate complex is initially constant (i.e. the steady-state assumption) yields the Michaelis–Menten approximation:

$$V = k_{\mathrm{cat}}[E]_{\mathrm{total}}\frac{[A]}{[A] + K_{\mathrm{M}}} \tag{5}$$

where V is the initial rate of the reaction, before appreciable changes occur in components' concentrations (equivalent to the initial rates of change in Eqs. (3) and (4); $[E]_{\mathrm{total}}$ is the total amount of E present in the system, i.e. $[E]_{\mathrm{total}}$ = $[E]$ + $[AE]$; and K_{M} = $(k_{\mathrm{r}} + k_{\mathrm{cat}})/k_{\mathrm{f}}$. The maximum reaction rate ($V_{\max}$ = $k_{\mathrm{cat}}[E]_{\mathrm{total}}$) is the rate under the extreme condition $[AE]$ = $[E]_{\mathrm{total}}$ and when $[A] \gg K_{\mathrm{M}}$.

The formulations above are all based on the well-stirred assumption, implying that there is a homogeneous distribution of molecules in space. If this is not reasonable, compartments may be introduced (e.g. intracellular/extracellular). Compartmentalization can be implemented by duplication of the relevant equations for the spatially segregated species, with a flux term where necessary, such as:

$$[A]_{\mathrm{ic}} \overset{C}{\longleftrightarrow} [A]_{\mathrm{ec}} \tag{6}$$

$$\frac{\mathrm{d}[A]_{\mathrm{ec}}}{\mathrm{d}t} = [C]\left(k_{\mathrm{f}}[A]_{\mathrm{ic}} - k_{\mathrm{r}}[A]_{\mathrm{ec}}\right) = -\frac{\mathrm{d}[A]_{\mathrm{ic}}}{\mathrm{d}t} \ , \tag{7}$$

where $[A]_{\mathrm{ic}}$ and $[A]_{\mathrm{ec}}$ are the concentrations of A in the intracellular and extracellular compartments, respectively, and C is a channel. Note that the rate constants here are in units of µM^{-1} s^{-1} and the reaction rate is again in μM s^{-1}. The equations are more complex, but well established [1], if membrane potential is involved.

If compartments are insufficient to describe the spatial distribution of the reactions, e.g. if one wishes to study the development of concentration gradients within a single compartment, then PDEs must be used in place of ODEs. In the resulting reaction–diffusion equation, rate of change of concentration varies with respect to several parameters (normally, two- or three-dimensional space and time) rather than one (time). For example, the equation describing reaction 2 would be:

$$\frac{\partial[B]}{\partial t} = D\nabla^2[B] + [AE]k_{\text{cat}} \,, \tag{8}$$

where the first term represents change due to diffusion, and the second term represents change due to reaction (*cf.* Eq. 4). D is the diffusion coefficient of B; $\nabla^2[B]$ is the three-dimensional Laplacian that describes the change in distribution of B in space, being the sum of the second derivatives of $[B]$ with respect to each spatial dimension. The equation's second term is simply the function that describes the reaction-driven rate of change of $[B]$ with respect to time [34].

As was alluded to above, one key assumption in the use of any deterministic model is that there are large numbers of molecules in the reactions. If this is not the case, then reality can deviate significantly from the predicted average behavior. Furthermore, ODE models also assume that the reactions take place in a "well-mixed" environment, i.e. concentrations are the same everywhere, as there is no representation of space. If this assumption is unreasonable, then the simplest solution, computationally, is to introduce compartments. For example, if a molecular species is capable of existing on either side of a membrane, it and its reactions must be represented twice – once for each compartment. If transit across the membrane is possible, then a flux term can be added (see Box 1).

Sometimes, independent compartments do not offer an appropriate solution for the representation of concentration differences in space, e.g. when studying chemical gradients across a single cellular compartment. In this case, PDEs can be used instead of ODEs. PDE models explicitly include spatial dimensions, in addition to time. The reaction–diffusion equation they utilize is detailed in Box 1; although its complete derivation is beyond the scope of this chapter, an excellent introduction to its physical meaning and application may be found in Ref. [6].

Deterministic modeling, with either ODEs or PDEs, requires the numerical solution of differential equations, using a solver. There are many algorithms available, whose precise workings are beyond the scope of this chapter. The modeler does not normally need to know the mechanistic details of a solver's function, although some understanding of its strengths and weaknesses can be useful. Numerical problems can lead to spurious simulation results and a modeler should be aware of what to look out for. Numerical problems are usually obvious, and inspection of the reaction rates and component concentrations over the time course of the simulation will reveal rapidly changing (spiking) or unreasonable or logically irrelevant values (unreasonable values may also be caused by errors in the model). In many cases, a simple operation such as reduction in the size of the time step will solve the problem. At other times, a change of algorithm is required; higher-order, variable time-step algorithms are usually recommended. Systems that have heterogeneous kinetics wherein some reactions are much faster than others are described as being *stiff*. The exact definition of stiffness has been investigated in considerable depth since the term was first proposed by Curtiss and Hirschfelder in 1952 [13]. Pragmatically, it refers to problems that can be solved better using implicit methods than explicit methods. These require an appropriate solver, which is included in many simulation packages, e.g. LSODA (http://www.llnl.gov/CASC/odepack) and Rosenbrock algorithms [12]. One drawback of a stiff solver is that, when solving nonstiff systems, it is slightly slower because of the extra calculations it carries out.

Depending on the size of the model and the time of the simulation, deterministic models can be fairly quick and easy to run, using, for example, MATLAB (http://www.mathworks.com) on a desktop PC. PDE models and large ODE models usually require many hours of processor time, however. One convenient option in these cases is the program Virtual Cell [57], which runs in a web browser and does all of its calculations on a centralized server of very powerful processors. This is not only faster than performing the calculations on the local machine, but allows the user to shut down their computer or run other software while the simulation progresses. Virtual Cell also has the benefit of an intuitively simple graphical user interface. Other alternatives to MATLAB and Virtual Cell include E-Cell [59], Gepasi/Copasi [45], Genesis/Kinetikit [62] and Berkeley Madonna.

Deterministic models have been used successfully to derive a number of insights about signaling systems. However, they are not always adequate or appropriate. They do not capture behavior that emerges from fluctuations, such as stochastic resonance, and they are only accurate when there are many molecules of each species involved in reactions.

2.2.2 Stochastic Models

Stochastic models deal with discrete molecules or reactions, rather than treating concentration as a continuous variable, and are probabilistic in nature. Stochastic models are frequently used in systems that are so physically small that concentration is no longer a meaningful parameter. For example, the distal part of a neuronal dendrite can be as little as 0.2 μm across; thus a 2-μm stretch of approximately cylindrical dendrite has a volume of just 6×10^{-17} L. If the concentration of a signaling molecule in that part of the neuron is, say, 100 nM, then there will be on average fewer than four molecules in the section of interest and deterministic modeling breaks down. Stochastic modeling may also be advantageous in physically larger systems when addressing the problem of very low concentration. A common example of this is the regulation of transcription. There are usually two copies of a gene in a eukaryotic cell, each having adjacent to it one or few binding sites for promoter sequences recognized by transcription factors. Again, concentration as a continuous variable lacks meaning in this situation. Thirdly, stochastic models are useful in situations where random fluctuations can determine system behavior. Recent work has shown, for example, that certain types of switching are stochastic in nature [48, 56] and there is a phenomenon known as stochastic resonance that produces oscillatory behavior based on the amplification of a low-level signal by random fluctuations. Stochastic resonance is a well-studied phenomenon and may be important in a range of biological processes, such as neuronal excitability. Its dynamical origins have been elucidated mathematically [2], so it can now be modeled deterministically, although it must be included specifically in the model design to do so.

The best-known stochastic algorithm for the simulation of biochemical kinetics comes from the work of Gillespie [21]. He developed a set of rules for calculating the probability of a given reaction occurring in a simulation time step, i.e. a *reaction probability density function*, based on kinetic rate parameters that can be measured physically. He showed that his Monte Carlo method relates directly to the chemical master equation, which describes the change over time of reactants' and products' existence-probability. The Gillespie algorithm is illustrated as a flow diagram in Figure 8. It is based on two assumptions – that the molecules are distributed evenly in space ("well mixed", as in a deterministic ODE model) and that the system is at thermodynamic (though not necessarily chemical) equilibrium. Since it was originally described, the Gillespie algorithm has been modified, improved and updated. In particular, Gibson developed the *next-reaction method*, which is a more efficient version of Gillespie's *direct method*, as well as techniques for conducting sensitivity analysis within a stochastic framework [19].

The Gillespie algorithm deals with reaction probabilities and so treats all molecules of a given species as identical. The alternative to this approach is an object-oriented algorithm, such as that developed by Morton-Firth and implemented in the program StochSim [49]. This algorithm tracks the number of molecules and complexes (i.e. objects) themselves, instead of reactions. When the absolute number of molecules is relatively small, but there are many

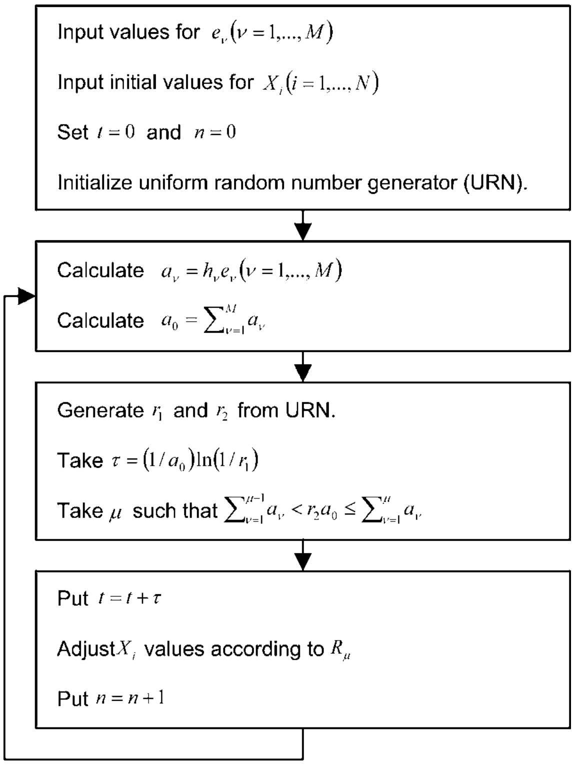

Input values for $e_\nu(\nu = 1,...,M)$
Input initial values for $X_i(i = 1,...,N)$
Set $t = 0$ and $n = 0$
Initialize uniform random number generator (URN).
Calculate $a_\nu = h_\nu e_\nu(\nu = 1,...,M)$
Calculate $a_0 = \sum_{\nu=1}^{M} a_\nu$
Generate r_1 and r_2 from URN.
Take $\tau = (1/a_0)\ln(1/r_1)$
Take μ such that $\sum_{\nu=1}^{\mu-1} a_\nu < r_2 a_0 \le \sum_{\nu=1}^{\mu} a_\nu$
Put $t = t + \tau$
AdjustX_i values according to R_μ
Put $n = n + 1$

Figure 8 Flow diagram for the Gillespie algorithm [21], where t is time and n is the number of steps taken; e_v and X_i are the reaction constants and concentrations, respectively, for the M reactions and N species in the simulation; e_v is a combinatorial factor describing the number of potential occurrences of a reaction in the current system state, usually calculated as product of the reactants concentrations; τ is the time to the next reaction (note that it is inversely proportional to the total reaction rate) derived from the reaction probability density function; and μs an index number that references a point on the probability density function, determining which reaction will take place. R_μ represents the reaction channel, i.e. the reaction that has just taken place, which determines the change in species concentration values, X_i. The algorithm is followed until some predetermined value of n or t is reached [21].

possible reactions and/or states for each molecule (the so-called *combinatorial explosion*), this technique can be significantly more efficient than the Gillespie algorithm. The latest releases of StochSim also have the advantage of being able to represent spatial heterogeneity.

Stochastic modeling, while more accurate in some circumstances than a deterministic approximation, requires a great deal of computer processor time. This is because stochastic simulations require an entire cycle of the algorithm to run for every single molecular-level reaction event. By comparison, each cycle of a deterministic solver's algorithm gives a reaction rate value, which typically would involve a large number of reacting molecules. For example, a 0.01-s simulation time step of the first-order reaction from Box 1, $AE \rightarrow B + E$, taking place in a cell of volume 1 pL, having a rate constant of 1 s^{-1} and an initial AE concentration of 100 nM, would proceed at an initial rate of 10^8 molecular reaction events per time step. Thus, if deterministic modeling is a valid option, it is usually by far the more efficient strategy. Nonetheless, with the increasing availability of affordable computing power and continuing improvements to stochastic calculation strategies, including multistep approximations [20] and even hardware solutions [55], stochastic simulation is fast becoming a practical approach to a wide range of problems.

As the Gillespie algorithm is based on an underlying master equation, it was proposed recently to solve the master equation by direct integration [38]. In this method, a set of deterministic ODEs is solved, where the dynamic variables of the equations are the probabilities of the system to be in the various states. This method is advantageous in several aspects: it provides temporal evolution of the concentrations which cannot be calculated by the Gillespie algorithm, and it is computationally efficient since no averaging is required and the master equation can be easily coupled to other deterministic rate equations in case a hybrid model is used.

2.2.3 **Hybrid Models**

As described above, deterministic and stochastic approaches to biochemical kinetic modeling have their own advantages and disadvantages. In order to exploit the benefits of both, it is possible to combine the two approaches in a so-called hybrid model. Parts of the system are represented deterministically and parts stochastically. In this way, one may take advantage of a deterministic model's computational efficiency (for those reactions where the necessary assumptions hold) and the stochastic model's accuracy (in representing low-concentration and low-volume effects, and fluctuations). For example, in our laboratory, we model the isotropic phase of fibroblast cell spreading. This process is driven by actin polymerization and thus a stochastic model is appropriate. However, the regulation of polymerization is by biochemical signaling reactions that can be modeled deterministically. In this case, custom C++ code has been written, which was necessary because of the particularly complex and dynamic geometry of the problem. For most modeling problems, however, programs are available that can perform hybrid modeling via a reasonably user-friendly interface. The most recent versions of Kinetikit (an add-on for the popular simulator, Genesis) have the capacity for hybrid modeling [61], as does the Biochemical Network Stochastic Simulator (BioNetS), which utilizes Gibson's enhancement of the Gillespie algorithm [1].

3 Identifying Parameters/Data Sets for Modeling

3.1 Functionally Relevant Connections

The cellular signaling network in eukaryotic cells is both vast and highly connected. Modeling the entire intracellular network is not currently feasible. The type of information sought, the size of the network being studied and the physical context in which the model operates determine the model type. This, in turn, limits the size of the network that can be modeled with the computational power available and determines the level of detail required to describe an interaction.

Modularity is a term that is commonly used to describe the structure of signaling functional modules. In engineering parlance, a module is a set of components that act together to perform a function, e.g. the navigation system in an airplane or an organ in an animal. When modeling signaling networks, in which many components are reused to a multitude of different ends, modules are defined relatively flexibly. Because of their context-dependence, it is often useful to think of functional modules. For example, PKA is involved in a large number of interactions and is therefore included in many mathematical models of signaling. However, the majority of interactions involving PKA are excluded from most models, because the inclusion of dozens of functionally unrelated actions can quickly make a system intractable. Thus, the boundaries of a module in a signaling network are defined by function. The decision about whether to include or exclude given interactions must be made on a case-by-case basis; clearly, the better informed one is about the functional importance of reactions, the better the decision. However, often there is inadequate information, and hence the need for modeling through trial and error must be employed, followed by careful validation.

A major repercussion of function-based component and connection selection is that models are invariably incomplete. It is therefore necessary to test a model against experimentally derived data in order to ensure that the omissions have not compromised realistic function and to examine which parameters warrant the greatest accuracy.

3.2 Qualitative Relationships

For graph theory analysis of networks and Boolean modeling, qualitative relationships are all the information that is needed. One must, however, be careful to include only data of acceptable quality. High-throughput techniques have produced a lot of data recently. The accuracy of the information content in these data sets is uncertain [29]. The yeast two-hybrid screen, for example, is a technique that involves the activation of a promoter by a split activator – one half of which is fused to the *prey* protein and the other to the *bait*. The fusion proteins are transfected into yeast, where, if they interact, the two halves of the activator come together and the reporter gene is transcribed [17]. Protein interaction networks for yeast and *Drosophila* have been constructed using this technology, but there was worryingly little overlap between some of the findings of different groups [27, 60] (see also Chapter 31). There are also theoretical objections to the application of this and some other high-throughput techniques to multicellular organisms. A protein–protein interaction is dependent on the simultaneous expression of both proteins in the appropriate subcellular compartment. Whether this occurs in the nucleus of a genetically manipulated yeast strain is of questionable relevance to metazoan biology. To avoid these issues we recommend to only select interactions identified by rigorous biochemical methods and verified by functional measurements. The limitation of this approach is that such rigorous described systems are incomplete. Additionally, for Boolean models, the type of interaction (stimulation/inhibition) must be known, and this is not available from any high-throughput interaction techniques; dynamic models necessitate still more accurate data.

3.3 Quantitative Specifications

Dynamic models of cell signaling networks ideally require experimentally measured catalytic and binding rates for the interactions they include. These can be measured by a range of enzymology techniques, which require expertise and a considerable investment of time. However, it is often possible to extract kinetic data from the biochemical literature. Sometimes direct measurements are available. In other cases, kinetic parameters may need to be inferred or fitted using a model. In both situations, accuracy and precision are not guaranteed, so a careful and critical appraisal of the published data is required. Some guidelines and recommendations are suggested here.

Terms such as $V_{\max}$ and k_{cat} are not always used in strict accordance with standard enzymology parlance. Care should be taken to assess that the methods used are consistent with the conclusions presented. Note that, even if nonstandard terminology is present, the appropriate numbers can often be extracted from the raw data.

When several papers describe a given interaction, apparently contradictory or inconsistent data may be reported and again one must be careful to examine the experimental methods used. In some cases, a simple factor such as temperature can account for the difference between reported values. A useful rule of thumb here is that rate constants tend to double for every 10 °C increase in temperature. This guideline is derived from the Arrhenius equation [35] and it holds only over the limited range of temperatures in which the interacting proteins are conformationally stable. Fortunately, in order to be meaningful, enzyme and binding kinetics should be measured in this appropriate range, so the rule is widely applicable.

In addition to temperature, other aspects of the experimental methodology can greatly affect the findings. Many kinetic parameters are measured *in vitro*, using purified proteins, which are often recombinant. In some cases, recombinant proteins display reduced enzyme activity or affinity for other proteins, due to missing or inappropriate post-translational modifications. For example, the affinity of rabbit liver $G\alpha_s$ for adenylylcyclase is around 100-fold greater than that of recombinant $G\alpha_s$ prepared from *Escherichia coli*. This occurs because $G\alpha_s$ derived from *E. coli* lacks a critical covalent modification [32]. *In vitro* measurements of enzyme activity may also be sensitive to parameters such as buffer composition (ion concentrations, pH, etc.). In some cases, enzymes are only partially purified for assay. These measurements do not always correspond to numbers derived from "purer" systems. The presence of unidentified accessory proteins, such as chaperones or scaffolds, may cause the observed kinetics to deviate from the "true" activity of the enzyme being studied. If the model does not contain explicit information about the influence of the accessory proteins – if they are unknown – then their presence in the experimental system may actually result in more realistic simulation results. Since a biochemical kinetic model is really just a mathematical representation of coupled input–output relationships, the precise details of the reaction mechanism may not be important. On the other hand, if the mechanistic details *are* of interest, then simulations can provide insight into the failings of the model, and what additional information should be sought experimentally.

Measurements of enzyme activity or binding affinity can also be made *in vivo*, e.g. by the construction of phosphorylation time courses on an immunoblot. These often require the fitting of data to a model, and, before using such results, a critical eye should be cast over the model and fitting technique that were used. One should also be careful when using data derived from populations of cells, because averaging effects can obscure the single-cell mechanism. Sometimes it is necessary to perform model fitting using raw published data. In this case, the freely available NIH image analysis program, ImageJ, is useful for quantifying responses from a graph, and programs such as Gepasi, Berkeley Madonna or MATLAB can be used to fit the extracted data.

There are a number of assumptions underlying the fitting of data to a function in order to extract useful parameters – principally involving the mechanism behind the observation. For example, association and dissociation data usually follow exponential curves, as these are the functions that result from the integration of the mass-action binding equations in Box 1. At times, the observations will be attributable to a more complex mechanism than what can be fitted. In this case, the modeler must decide whether a black box simplification is acceptable or whether experiments should be conducted to investigate the details of interactions.

In addition to kinetic parameters, quantitative modeling requires initial or total concentrations of signaling molecules. In some cases, as with kinetic parameters, these have been estimated directly and can be found in the literature. In other cases, biochemical characterization studies may be amenable to back-calculation of concentration from protein standard curves, enzyme purification tables and cell numbers used (usually one must assume a cell volume, which should be estimated according to cell type). In some cases, one can simulate an enzymatic reaction using Michaelis–Menten kinetics, so concentration and k_{cat} as individual parameters are not even necessary. Enzyme purification and activity tables frequently give the "fold" purification (*FP*; no units) and enzyme activity (*A*; typically expressed in μmol min^{-1} mg^{-1}); these, combined with an assumed protein yield and a few units-conversions, give the V_{max} in μM s^{-1}.

At times, it is not feasible to derive model parameters of interest from the experimental literature and it may be almost impossible to access it experimentally. In these cases, the kinetics of homologous enzymes, or the affinities of proteins with identical interacting motifs, may be used as a starting point for parameter estimation. Thus estimates are then validated by comparing the systems-level behavior of the model with an appropriate experimental assay.

4 Model Validation

4.1 Parameter Variation and Sensitivity Analysis

Once a model has been built, it should be validated. As discussed above, some parameters may not be as critical as others or there may be potentially important connections that were excluded for pragmatic reasons. The principal means of testing whether the behavior of the model system is realistic is through comparison with experimental observations. Time courses for the activation of various model components are often measurable. Phosphorylation, which is commonly an activation step at some point in the system (in cell signaling, usually several), can be detected using phospho-specific antibodies, ^{32}P incorporation or by looking at molecular weight shifts in protein immunoblots. It is preferable to conduct such experiments in-house, in order to control for the variation in experimental factors that can affect response reproducibility, but it may be possible to find the necessary data in the literature. Essentially, this is similar to model fitting and some parameters may need to be "tweaked" (varied by a small amount) in order to reproduce the experimental findings. This differs from the quantification of kinetics by model fitting, which is used to find parameters for the model in its construction. Time course comparisons, by contrast, compare effects that are produced by combinations of many parameters, as is typically of interest in modeling a biological problem.

In addition to testing the conformity of the model with experimental data, parameter variation can be used to identify which parameters are the main determinants of system behavior. Ideally, one should aim to have reliable estimates of these key parameters, whereas less accuracy may be sufficient for noncritical parameters. A more formal type of parameter variation may also be performed, using sensitivity analysis. There are many sophisticated techniques for performing this analysis [22], but all estimate the sensitivity of the output to variations in a particular input variable. Sensitivity analysis is typically sampling based – an algorithm performs the simulation several times with automatic variation across a random or assumed distribution of an input parameter and assesses the effect on the output parameter. It is particularly useful if one is interested in a single, well-defined, output parameter, but a more informal approach to parameter variation can also be very informative if one wishes to obtain a mechanistic understanding of the system as a whole.

4.2 Constraints and Predictions

In some situations, there is inadequate information to model a system with precise mechanistic accuracy. During the process of validation, one may identify an input–output relationship in the model that is behaving unrealistically. One must then decide whether the model needs refining by experimental investigation, i.e. whether the components and reactions in the problematic region are of particular relevance or interest. If not, or if the correct mechanism is experimentally inaccessible, it is possible to constrain the model at that particular level using experimentally derived input–output curves, essentially putting a small "black box" into a peripheral part of the system. In this way, technical limitations need not prevent insight into system dynamics.

Mathematical modeling should be able to make useful predictions that cannot be surmised intuitively. A major advantage to having a model is the ease with which it can be manipulated, and this can yield insights into complex system dynamics. Thus, one can identify key parameters or components that control network behavior. These predictions can then be tested by genetic manipulations that change the concentration of cellular components, and with pharmacological inhibitors of enzymes, which change their kinetic parameters (k_{cat} and/or K_{M}). One long-range goal is to identify network control points that may be utilized to manipulate signaling networks in a clinical setting, to diagnose or treat disease.

5 Perspective

The signaling network connects functionally dedicated cellular “machines”, not only by passing information across the cell via signaling pathways, but also processing information as it traverse the network by network connectivity that results in regulatory motifs. In this way, the system context is taken into account to ensure an appropriate response. In some cases, the signaling system has evolved a limited range of responses and so “switches” between predetermined states. In other cases, signaling is analog, conferring response flexibility.

Signaling systems are undeniably complex, but their elucidation will no doubt prove highly valuable in the understanding and treatment of disease. In many cases, the number of permutations to be investigated experimentally is daunting. Mathematical modeling can offer a formal and convenient approach for discovering systems-level behavior; for determining the roles not only of signaling molecules, but of their “circuitry”. Many insights revealed by modeling are nonintuitive to the biologist and would therefore be difficult to arrive at without a mathematical description of the system. Of course, mathematical predictions need to be confirmed experimentally, but modeling enables us to choose the right types of experiment to understand systems-level behaviors. All understanding is, in a sense, modeling. Textbooks and reviews are replete with cartoons of mechanisms. Implicit in the derivation of these from raw laboratory data is the implementation of a model. The use of mathematics to make explicit predictions constrains the modeling process strictly within the bounds of logic. Specifically, in the field of cell signaling, biochemical kinetic modeling is the gold standard for making predictions. Deterministic modeling is currently the standard approach, while stochastic modeling is often used in situations where the deterministic assumptions are violated.

The technological strides of the last decades are being felt in all realms of biology. Thus, as techniques for experimental measurement improve, the need for more penetrating analysis grows. No doubt the greatest insights into cell signaling will come from advances in systems-level analysis, in which mathematical modeling must necessarily play a leading role.

Acknowledgments

We gratefully acknowledge the valuable comments and advice of Narat J. Eungdamrong and Yuguang Xiong. Research in the Iyengar laboratory is supported by NIH grants GM-54508 and GM-72853. A. M. was supported by the Pharmacological Science training grant from NIGMS (GM-62754).

Glossary

Bistability/multistability The existence of two (bistability) or more (multistability) steady states of activity for a system. This can confer switching behavior, comparable with analog-to-digital signal conversion.

CaMK I–IV Calmodulin-binding protein kinases that require activated calmodulin for their kinase activity.

Coherent/incoherent feedforward loop A feedforward loop is a network motif made up of three or more nodes (see Section 1.2.3) arranged such that there is one input and one output node. These nodes are connected through two separate cascades where the signal splits at the input node and converges at the output node. The feedforward loop is said to be coherent if these two "arms" have the same overall sign (positive or negative) and incoherent if the arms differ qualitatively in their influence on the output.

Compartmental model A model in which space is represented as compartments – discrete mathematical systems, sometimes connected by fluxes of components. Normally, ordinary differential equations are used to represent the reactions in and between the compartments.

Cybernetics A field of science dealing with communication and control in nature and in man-made engineered systems.

Diacylglycerol (DAG) A second messenger generated by the hydrolysis of inositol phospholipids typically by the enzyme phospholipase C. Due to its hydrophobic properties, DAG is in the plasma membrane.

Deterministic model The representation of reactions by rate equations (ordinary differential equations or partial differential equations) such that the outcome of running the model is determined absolutely by the input variables, with no random fluctuations or chance occurrences.

Guanine nucleotide exchange factor (GEF) A regulator that catalyzes the release of guanosine diphosphate, permitting guanosine triphosphate to bind.

GTPase-activating protein (GAP) A protein that stimulates the hydrolysis of guanosine triphosphate to guanosine diphosphate.

Hybrid model The representation of a reaction system by a combination of deterministic and stochastic reaction equations.

Mitogen-activated protein kinase (MAPK) A family of ubiquitous kinase enzymes, regulated by phosphorylation, which control many important cellular functions, such as gene transcription.

Network motif A characteristic pattern of connections between network components that have the potential to perform information processing functions and are statistically enriched in network abstraction of real-world complex systems. Examples include feedback and feedforward loops.

Ordinary differential equation (ODE) A rate equation that describes the change in dependent variables (in biochemical kinetic modeling, concentrations) with respect to one independent variable (usually time).

Partial differential equation (PDE) A rate equation that describes the change of dependent variables (usually concentrations) with respect to two or more independent variables (usually time plus one to three spatial dimensions).

Protein kinase A (PKA or cAMP-dependent protein kinase) A ubiquitous signaling enzyme that has serine/threonine kinase activity. PKA modulates the activity of its targets by covalently attaching a phosphate group to its substrate. PKA phosphorylates a range of targets, depending on cell type, in response to elevated intracellular concentrations of cAMP (adenosine $3',5'$-cyclic monophosphate).

Protein kinase B (PKB or Akt) Serine/threonine protein kinase which mostly acts as a promoter for survival signals. The kinase acts downstream of lipid signaling pathways, e.g. in the insulin receptor signal transduction pathway.

Protein kinase C (PKC) A family of protein serine/threonine kinases divided into two subfamilies: conventional and atypical. Conventional PKCs require calcium and diacylglycerol, and are activated through the Gq and phospholipase C pathway.

Protein kinase D (PKD) Calcium-independent diacylglycerol-dependent serine/threonine kinase; belongs to the PKC family of protein kinases.

Protein kinase G (PKG or cGMP-dependent kinase) A serine/threonine protein kinase activated by cGMP and mostly known for its function in regulating smooth muscle relaxation.

Stochastic model A model in which reactions are simulated according to probability of occurrence rather than rates.

References

1 Adalsteinsson, D., D. McMillen and T. C. Elston. 2004. Biochemical Network Stochastic Simulator (BioNetS): software for stochastic modeling of biochemical networks. BMC Bioinformatics **5**: 24.

2 Arai, K., S. Mizutani and K. Yoshimura. 2004. Deterministic stochastic resonance in a Rossler oscillator. Phys. Rev. E **69**: 026203.

3 Arkin, A. P. 2000. Signal processing in biochemical reaction networks. In Walleczek, J. (ed.), *Self-Organized Biological Dynamics and Nonlinear Control: Toward Understanding Complexity, Chaos and Emergent Function in Living Systems*. Cambridge University Press, Cambridge: 112–44.

4 Barabasi, A.-L. and R. Albert. 1999. Emergence of scaling in random networks. Science **286**: 509–12.

5 Barrios-Rodiles, M., K. R. Brown, B. Ozdamar, et al. 2005. High-throughput mapping of a dynamic signaling network in mammalian cells. Science **307**: 1621–5.

6 Berg, H. C. 1993. *Random Walks in Biology*, expanded edn. Princeton University Press, Princeton, NJ.

7 Bhalla, U. S. and R. Iyengar. 1999. Emergent properties of networks of biological signaling pathways. Science **283**: 381–7.

8 Bhalla, U. S., P. T. Ram and R. Iyengar. 2002. MAP kinase phosphatase as a locus of flexibility in a mitogen-activated protein kinase signaling network. Science **297**: 1018–23.

9 Bornholdt, S. 2005. Systems biology: less is more in modeling large genetic networks. Science **310**: 449–51.

10 Carlson, J. M. and J. Doyle. 1999. Highly optimized tolerance: a mechanism for power laws in designed systems. Phys. Rev. E **60**: 1412–27.

11 Chaves, M., R. Albert and E. D. Sontag. 2005. Robustness and fragility of Boolean models for genetic regulatory networks. J. Theor. Biol. **235**: 431–49.

12 Corana, A., M. Marchesi, C. Martini and S. Ridella. 1987. Minimizing multimodal functions of continuous variables with the "simulated annealing" algorithm. ACM Trans. Math. Software **13**: 272–80.

13 Curtiss, C. F. and J. O. Hirschfelder. 1952. Integration of stiff equations. Proc. Natl Acad. Sci. USA **38**: 235–43.

14 Dekel, E., S. Mangan and U. Alon. 2004. Environmental selection of the feed-forward loop circuit in gene-regulation networks. Phys. Biol. **2**: 81–8.

15 Eungdamrong, N. J. and R. Iyengar. 2004. Computational approaches for modeling regulatory cellular networks. Trends Cell Biol. **14**: 661–9.

16 Ferrell, J. E., Jr. 2002. Self-perpetuating states in signal transduction: positive feedback, double-negative feedback and bistability. Curr. Opin. Cell Biol. **14**: 140–8.

17 Fields, S. and O.-k. Song. 1989. A novel genetic system to detect protein–protein interactions. Nature **340**: 245–6.

18 Friedman, N., M. Linial, I., Nachman and D. Pe'er. 2000. Using Bayesian networks to analyze expression data. J. Comput. Biol. **7**: 601–20.

19 Gibson, M. A. 2000. *Computational Methods for Stochastic Biological Systems*. California Institute of Technology, Pasadena, CA.

20 Gillespie, D. T. 2001. Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys. **115**: 1716–33.

21 Gillespie, D. T. 1977. Exact stochastic simulation of coupled chemical reactions. J. Phys. Chem. **81**: 2340–61.

22 Helton, J. C., F. J. Davis and J. D. Johnson. 2005. A comparison of uncertainty and sensitivity analysis results obtained with random and Latin hypercube sampling. Reliab. Eng. Syst. Saf. **89**: 305–30.

23 Hokin, M. R. and L. E. Hokin. 1953. Enzyme secretion and the incorporation of P32 into phospholipids of pancreas slices. J. Biol. Chem. **203**: 967–77.

24 Holland, J. H. 2002. Exploring the evolution of complexity in signaling networks. Complexity **7**: 34.

25 Huang, S. and D. E. Ingber. 2005. Cell tension, matrix mechanics, and cancer development. Cancer Cell **8**: 175–6.

26 Irish, J. M., R. Hovland, P. O. Krutzik, O. D. Perez, O. Bruserud, B. T. Gjertsen and G. P. Nolan. 2004. Single cell profiling of potentiated phospho-protein networks in cancer cells. Cell **118**: 217–28.

27 Ito, T., T. Chiba, R. Ozawa, M. Yoshida, M. Hattori and Y. Sakaki. 2001. A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proc. Natl Acad. Sci. USA **98**: 4569–74.

28 Janes, K. A., J. G. Albeck, S. Gaudet, P. K. Sorger, D. A. Lauffenburger and M. B. Yaffe. 2005. A systems model of signaling identifies a molecular basis set for cytokine-induced apoptosis. Science **310**: 1646–53.

29 Jansen, R. and M. Gerstein. 2004. Analyzing protein function on a genomic scale: the importance of gold-standard positives and negatives for network prediction. Curr. Opin. Microbiol. **7**: 535–45.

30 Jenster, G. 2004. A visualisation concept of dynamic signalling networks. Mol. Cell. Endocrinol. **218**: 1–6.

31 Kauffman, S. A. 1993. *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, Oxford.

32 Kleuss, C. and A. G. Gilman. 1997. Gsalpha contains an unidentified covalent modification that increases its affinity for adenylyl cyclase. Proc. Natl Acad. Sci. USA **94**: 6116–20.

33 Kuipers, B. 1994. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, Cambridge, MA.

34 Kuyucak, S. and S.-H. Chung. 2002. Permeation models and structure–function relationships in ion channels. J. Biol. Phys. **28**: 289–308.

35 Laidler, K. J. 1993. *The World of Physical Chemistry*. Oxford University Press, Oxford.

36 Lauffenburger, D. A. 2000. Cell signaling pathways as control modules: complexity for simplicity? Proc. Natl Acad. Sci. USA **97**: 5031–3.

37 Laurent, M. and N. Kellershohn. 1999. Multistability: a major means of differentiation and evolution in biological systems. Trends Biochem. Sci. **24**: 418–22.

38 Lipshtat, A., H. B. Perets, N. Q. Balaban and O. Biham. 2005. Modeling of negative autoregulated genetic networks in single cells. Gene **347**: 265–71.

39 Luscombe, N. M., M. M. Babu, H. Yu, M. Snyder, S. A. Teichmann and M. Gerstein. 2004. Genomic analysis of regulatory network dynamics reveals large topological changes. Nature **431**: 308–12.

40 Ma'ayan, A., S. L. Jenkins, S. Neves, et al. 2005. Formation of regulatory patterns during signal propagation in a mammalian cellular network. Science **309**: 1078–83.

41 Mangan, S. and U. Alon. 2003. Structure and function of the feed-forward loop network motif. Proc. Natl Acad. Sci. USA **100**: 11980–5.

42 Mangan, S., S. Itzkovitz, A. Zaslaver and U. Alon. 2006. The incoherent feed-forward loop accelerates the response-time of the gal system of *Escherichia coli*. J. Mol. Biol. **356**: 1073–81.

43 Mangan, S., A. Zaslaver and U. Alon. 2003. The coherent feedforward loop serves as a sign-sensitive delay element in transcription networks. J. Mol. Biol. **334**: 197–204.

44 Manning, G., G. D. Plowman, T. Hunter and S. Sudarsanam. 2002. Evolution of protein kinase signaling from yeast to man. Trends Biochem. Sci. **27**: 514–20.

45 Mendes, P. 1993. GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. Comput. Appl. Biosci. **9**: 563–71.

46 Michaelis, L., Menten, M. L. 1913. Die Kinetik der Invertinwirkung. Biochem. Z. **49**: 333.

47 Middendorf, M., E. Ziv and C. H. Wiggins. 2005. Inferring network mechanisms: the *Drosophila melanogaster* protein interaction network. Proc. Natl Acad. Sci. USA **102**: 3192–7.

48 Miller, P., A. M. Zhabotinsky, J. E. Lisman and X. J. Wang. 2005. The stability of a stochastic CaMKII switch: dependence on the number of enzyme molecules and protein turnover. PLoS Biol. **3**: e107.

49 Morton-Firth, C. J. and D. Bray. 1998. Predicting temporal fluctuations in an intracellular signalling pathway. J. Theor. Biol. **192**: 117–28.

50 Murphy, L. O., J. P. MacKeigan and J. Blenis. 2004. A network of immediate early gene products propagates subtle differences in mitogen-activated protein kinase signal amplitude and duration. Mol. Cell. Biol. **24**: 144–153.

51 Prudhomme, W., G. Q. Daley, P. Zandstra and D. A. Lauffenburger. 2004. Multivariate proteomic analysis of murine embryonic stem cell self-renewal versus differentiation signaling. Proc. Natl Acad. Sci. USA **101**: 2900–5.

52 Rodbell, M. 1995. Nobel Lecture: Signal transduction: evolution of an idea. Biosci. Rep. **15**: 117–33.

53 Sachs, K., D. Gifford, T. Jaakkola, P. Sorger and D. A. Lauffenburger. 2002. Bayesian network approach to cell signaling pathway modeling. Sci. STKE **2002**: pe38.

54 Sachs, K., O. Perez, D. Pe'er, D. A. Lauffenburger and G. P. Nolan. 2005. Causal protein-signaling networks derived from multiparameter single-cell data. Science **308**: 523–9.

55 Salwinski, L. and D. Eisenberg. 2004. *In silico* simulation of biological network dynamics. Nat. Biotechnol. **22**: 1017–9.

56 Samoilov, M., A. P. Arkin and J. Ross. 2002. Signal processing by simple chemical systems. J. Phys. Chem. A **106**: 10205–21.

57 Schaff, J., C. C. Fink, B. Slepchenko, J. H. Carson and L. M. Loew. 1997. A general computational framework for modeling cellular structure and function. Biophys. J. **73**: 1135–46.

58 Sutherland, E. W. and T. W. Rall. 1958. Fractionation and characterization of a cyclic adenine ribonucleotide formed by tissue particles. J. Biol. Chem. **232**: 1077–1091.

59 Tomita, M., K. Hashimoto, K. Takahashi, et al. 1999. E-CELL: software environment for whole-cell simulation. Bioinformatics **15**: 72–84.

60 Uetz, P., L. Giot, G. Cagney, et al. 2000. A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. Nature **403**: 623–7.

61 Vasudeva, K. and U. S. Bhalla. 2004. Adaptive stochastic-deterministic chemical kinetic simulations. Bioinformatics **20**: 78–84.

62 Vayttaden, S. J. and U. S. Bhalla. 2004. Developing complex signaling models using GENESIS/Kinetikit. Sci. STKE **2004**: pl4.

63 von Mering, C., R. Krause, B. Snel, M. Cornell, S. G. Oliver, S. Fields and P. Bork. 2002. Comparative assessment of large-scale data sets of protein–protein interactions. Nature **417**: 399–403.

64 Watts, D. J. and S. H. Strogatz. 1998. Collective dynamics of "small-world" networks. Nature **393**: 440–2.

65 Wolf, D. M. and A. P. Arkin. 2003. Motifs, modules and games in bacteria. Curr. Opin. Microbiol. **6**: 125–34.

66 Yang, D., S. O. Zakharkin, G. P. Page, J. P. Brand, J. W.Edwards, A. A.Bartolucci and D. B. Allison. 2004. Applications of Bayesian statistical methods in microarray data analysis. Am. J. PharmacoGenomics **4**: 53–62.

23 Dynamics of Virus–Host Cell Interaction

Udo Reichl and Yury Sidorenko

1 Introduction

For little more than 100 years viruses have been known as small, infectious, obligate intracellular parasites, which efficiently use a wide range of host cells for their reproduction. It was not until the early 20th century when their structural simplicity was revealed by electron microscopy and a first rational classification of viruses based on their morphology was proposed. Over the following decades steady technical advances and enormous progress in molecular biology not only allowed to precisely define structure, molecular composition and synthesis of virus particles, but also to investigate their relationship with their host cells.

The main interest in viruses derives from the understanding of their role as agents responsible for epidemics of contagious diseases. Reports on viral diseases can be found in numerous ancient records, e.g. rabies, polio, smallpox and, probably, influenza. From the earliest times efforts have been undertaken to understand viral pathogenesis, and to prevent and control epidemics. Today, the main focus of academic and pharmaceutical research is on the identification and molecular biology of new and emerging viruses such as human immunodeficiency virus (HIV), hepatitis virus or influenza virus. In addition, enormous efforts are undertaken to prevent viral diseases by vaccines and to develop new vaccine technologies and antiviral drugs.

Viral pathogenesis and virus spreading can be analyzed on several levels (Figure 1), which comprise

- Structural aspects and detailed molecular level interactions of virus and host cell components.
- The interaction of virions with a single host cell.
- The spreading of infectious agents within populations of cells, tissues and organs.
- The infection of an individual resulting in an immune defense.
- The establishment of a viral disease in a host population.

Bioinformatics - From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

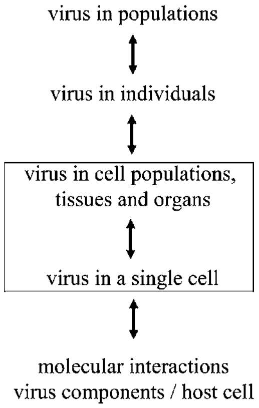

virus in populations
virus in individuals
virus in cell populations, tissues and organs
virus in a single cell
molecular interactions virus components / host cell

Figure 1 Modeling hierarchy. Virus dynamics can be investigated at different levels ranging from the detailed description of molecular mechanisms relevant for the interaction of viral components with their host cells (e.g. conformational changes of the influenza surface protein HA required for the fusion of viral and host cell membrane) to the spreading of virions in individuals (micro-epidemiology [44]) and populations (traditional epidemiology). Inner box: levels discussed in this chapter.

This chapter focuses on the mathematical modeling of virus replication in single cells and the (intracellular) interaction of virus components with their host cell (structured, unsegregated models; see Section 3). Aspects covered are virus attachment and entry, transcription and translation of viral mRNA, genome replication, assembly, and virus release. Additionally, aspects of host cell defenses, interference with cellular signal transduction networks and the impact of infections on host cell gene expression are addressed. Many of the experimental results required for these models have been derived from one-step growth experiments [10] in static cultures, shake flasks and bioreactors under controlled cultivation conditions where populations of cells are infected at different multiplicity of infection (m.o.i., i.e. the number of infectious units per cell). Therefore, mathematical models describing virus spreading from cell to cell are also discussed briefly (unstructured, unsegregated models; see Section 3).

Mathematical models for intracellular virus–host cell interactions have been developed for a limited number of viruses including bacteriophages, baculoviruses, HIV, Semliki Forest virus and influenza A virus [7, 13, 25, 48, 50, 52]. Most of the existing models address only specific aspects of virus replication such as virus binding, endocytosis or viral RNA synthesis. So far, the impact of viral infections on mammalian cells with regard to their signal transduction processes, genome expression or apoptosis is being investigated mainly experimentally. No attempt has been made to include these results obtained for eukaryotic cells into existing models, e.g. to develop mathematical models, which include virus attachment and viral genome replication as well as aspects of virus-induced apoptosis. However, to cope with the increasing amount of quantitative and semiquantitative data obtained from infection studies and to organize this information into a coherent whole the rigorous use of mathematical modeling and the development of appropriate modeling and simulations tools seems to be indispensable. Therefore, the existing models (even with their limitations and drawbacks) should be considered as a first step towards a quantitative and integrative understanding of virus–host cell interaction and a key factor to establish a systems biology platform for virus dynamics. This platform should allow:

- Simulating virus replication on a cellular level.
- Investigating requirements and limitations of viral growth in a host cell.
- Identifying targets for antiviral compounds.
- Evaluating new therapeutic strategies.
- Optimizing viral-based production systems.

2 Viral Infection of Cells

According to the latest report of the International Committee on Taxonomy of Viruses (ICTV) [4] there are about 3000 identified virus species that infect bacteria, fungi, plants, animal and human cells. The classification of these viruses is based on their shared properties [15], which include the nature of the nucleic acid (DNA, RNA), the symmetry of the capsid, the presence or absence of an envelope and their size (virion diameter, genome length). As obligate intracellular parasites, all viruses have to undergo a series of steps for the successful completion of their life cycle inside their host cell:

- Cell attachment
- Internalization
- Release of viral genome
- Decoding of viral genome information
- Viral protein synthesis
- Viral genome replication
- Virus assembly
- Release of newly produced virions

In addition, the transport of viral genome into the nucleus (most DNA viruses, retroviruses, influenza viruses, etc.), the supply of cellular precursors for viral genome replication and protein synthesis, the intracellular transport of viral components as well as host cell-specific defense mechanisms have to be considered.

Typically, the one-step growth cycle of viruses is experimentally studied in cell cultures under defined conditions. Therefore, their host cells are grown in suspension or monolayer cultures to their optimal growth phase (late logarithmic phase). After the removal of the cell culture medium the virus seed is added with a m.o.i. $>$ 5–10 to guarantee rapid infection of all cells. In a next step the cells are washed to remove any unabsorbed virions, fresh medium is added and the kinetics of virus production is monitored. As an example a one-step growth curve for lytic bacteriophages in *Escherichia coli* is shown in Figure 2. It starts with a lag phase (eclipse period) during which attachment, uncoating, phage synthesis and the intracellular formation of the first complete phages takes place. During this period of time no infectious particles can be detected in the cell culture supernatant (except for a more or less low constant level of infectivity resulting from phages which only attached to their host cells without internalization). After a brief period of exponential increase the number of intracellular infectious particles increases at a constant rate (rise period) due to limitations in the supply of cellular precursors or other limitations in bacterial synthesis capacity. Eventually, metabolism and cell structure of infected bacteria breaks down, and progeny phages are released into the extracellular medium. The burst size corresponds to the number of progeny phages produced per cell. In the following two subsections viral infections in prokaryotic and eukaryotic cells are discussed in more detail.

2.1 Viral Infection of Prokaryotic Cells

Since the 1930s, when the life cycle of bacteriophages in cell cultures was first quantitatively analyzed in *Staphylococcus aureus* [28] and *E. coli* [10], enormous progress has been achieved towards the understanding of bacteriophage–host cell interaction. Mainly, this is due to the vast amount of information available on the molecular biology of the host cells. In particular, for *E. coli*, which was selected as one of the model organisms in systems biology [14], an extensive set of data on macromolecular composition, cellular synthesis rates, signal transduction, gene expression, etc., is available (e.g. Ref. [3]). In addition, bacteria allow for detailed phage growth experiments under well-defined conditions in shake flasks and bioreactors to investigate phage replication with respect to the physiological status of the host cell (e.g. Refs. [23, 64]).

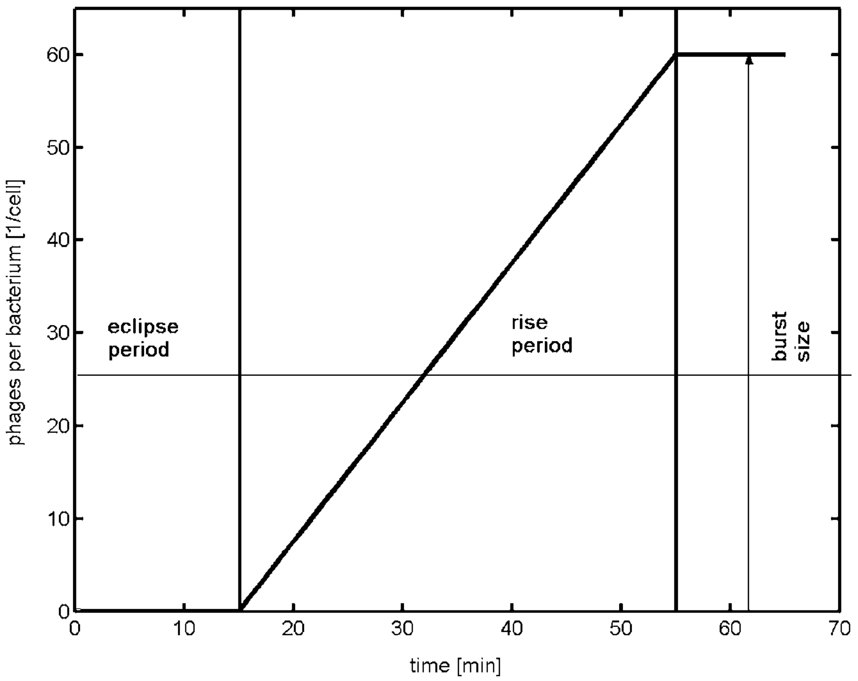


Figure 2 Schematic: intracellular one-step growth cycle of bacteriophages in *E. coli*. After about 15 min p.i. (eclipse period) the intracellular number of progeny phages increases (rise period; assumed rise rate α = 1.5 phages min^{-1}) until phage synthesis ceases due to lysis of the host cell. Here, the burst size is about 60 phages cell^{-1}.

Consequently, mathematical models that describe the intracellular growth cycle of bacteriophages have been developed for more than 15 years at various levels of complexity. Rabinovitch and coworkers [48], for instance, show that simple mathematical relations can be used to describe one-step growth experiments for bacteriophage T4 development in *E. coli* (see Section 3.1). In addition, the influence of bacterial growth rate on phage development [47] and the kinetics of bacterial lysis by phages are investigated in detail in Ref. [49]. In contrast, highly structured models (see Section 3.1) for the intracellular growth of other bacteriophages (T3/T7) in *E. coli* have been developed by various research groups (e.g. Refs. [5,13,65]). With such models a first attempt has been made to quantitatively understand phage replication at a molecular level. Based on the extensive set of data available for the genetics, physiology and macromolecular composition of host cells, aspects of phage gene replication, phage protein synthesis, phage assembly and its dependence on host physiology were investigated. Other mathematical models have been developed to describe the use of intracellular resources and energy consumption for the growth of phage Qβ in *E. coli* [27]. Here, it could be shown that protein synthesis of phages, particularly translation at the ribosomes, dominates phage growth. This observation is in agreement with the energy requirements of exponentially growing bacteria [56] as well as with the influence of ribosome number and protein synthesis rate per ribosome on the growth rate of exponential cultures [3]. Furthermore, it could be shown that phages efficiently use host cell resources and that overall yields are comparatively insensitive to changes in model parameters such as RNA-to-ribosome or RNA-to-replicase binding.

2.2 Viral Infection of Eukaryotic Cells

Research on growth characteristics and genetic properties of viruses infecting eukaryotic cells dates back into early 1950s when Dulbecco analyzed plaques of western equine encephalomyelitis virus in primary chicken fibroblast cultures [9]. Over the following years continuous improvements in the cultivation of cells, the generation of immortal cell lines and advances in assay techniques allowed us to investigate the life cycle of plant, animal and human viruses under precisely controlled conditions in analogy to the one-step growth cycle experiments of bacteriophages. Since then, our understanding of the molecular basis of virus–host cell interaction in eukaryotic cells has improved considerably. In particular, viruses infecting animals and humans have attracted enormous research efforts to prevent and treat viral infections as well as to avoid the spread of viral diseases. Today, highly efficacious vaccines are available against a number of viral diseases such as smallpox, polio, influenza or hepatitis A. In addition, new vaccine technologies, e.g. viral vectors and DNA vaccines, are being developed to complement existing vaccines or as potential candidates for HIV vaccines (see Chapter 39). Eventually, research concerning the molecular mechanisms of virus–host cell interaction has become a prerequisite for the development of antiviral drugs and serves as a powerful tool for understanding fundamental aspects of cellular functions.

So far, most mathematical models which describe the intracellular virus life cycle in eukaryotic cells focus either on the initial steps of the virus infection cycle or omit some intermediate steps of the replication cycle such as transport processes across cell compartments or turnover of cellular pools of precursors required for viral genome and protein synthesis. Mainly, these models have been developed for biotechnology and biomedical applications.

In biotechnology, the majority of these mathematical models have been developed to describe dynamic aspects relevant for the production of viruses or recombinant viral proteins in cell cultures. A model from Dee and coworkers in 1995 [7] for the early steps of the infection cycle of Semliki Forest virus considers the initial steps of virus trafficking in baby hamster kidney cells (BHK-21). In addition to the binding of the viruses to the cell surface receptors, endocytosis and RNA synthesis are taken into account. However, neither viral protein synthesis nor assembly or release of progeny virus particles are considered. A similar model was used later by the same group to investigate options to design infection strategies for baculovirus production in insect cell cultures [8]. In a further modeling approach baculovirus replication in insect cells for the manufacturing of recombinant proteins was investigated by Jang and coworkers in 2000 [25]. The model considers cellular metabolism and key features of the infection cycle of baculovirus, but focuses mainly on recombinant protein production. Therefore, attachment kinetics, cellular transport or available pools of amino acids, nucleotides and energy requirements for virus synthesis are not addressed.

Mathematical models of intracellular virus growth can also be useful for biomedical applications, e.g. to support the evaluation of anti-HIV strategies or to improve our understanding of virus-mediated diseases on a cellular level. Based on the extensive information available on molecular aspects of HIV-1 replication, Reddy and Yin [50] have formulated a highly structured model of the virus growth cycle in CD4$^+$ T lymphocytes. The model covers most of the key features of the replication cycle from the reverse transcription of the viral genome and its integration into the host genome to virus budding in maturation. Based on the model it was not only possible to quantitatively describe the dynamics of transcription, translation and virion production, but also to investigate the influence of potential drug targets for anti-HIV therapies.

In contrast to the replication of phages in bacteria, a quantitative understanding of the interaction of viruses with eukaryotic cells is more difficult to obtain for a number of reasons. First, cellular growth, gene expression, protein synthesis, transport processes and metabolism in eukaryotes are far more complex compared to bacteria. Second, eukaryotic cells are highly structured containing various specialized compartments relevant for virus replication, e.g. the different compartments of the secretory pathway required for covalent modifications, transport and apical sorting of influenza virus membrane proteins. Third, several cellular key processes required for a comprehensive understanding of virus–host cell interaction are the subject of current research activities and are not elucidated in sufficient detail to be used in mathematical models for virus dynamics. These include cellular signal transduction processes, in particular the specific interaction of viruses or viral proteins with signal transduction networks of their host cell, and specific and unspecific defense mechanisms of host cells, such as the production of interferon and the impact of viral gene products on programmed cell death (apoptosis) of infected cells [35, 36]. Fourth, comprehensive data sets (and mathematical models) available for growth and physiological status of "model organisms" such as *E. coli* are not available for higher cells, except for the yeasts *Saccharomyces cerevisiae* and *Schizosaccharomyces pombe* [14]. So far, no animal or human "model cell" has been selected and thoroughly characterized with respect to general aspects of its structure, growth, metabolism and, specifically, virus–host cell interaction. For example, while the first steps have been made to develop detailed mathematical models for the replication of HIV in $CD4^+$ T lymphocytes [50] or influenza A virus in animal cells [52], most kinetic parameters, synthesis and degradation rates as well as quantitative information on cellular composition had to be taken from numerous publications and textbooks. An additional problem is the fact that present data is derived from experiments performed under various cultivation conditions, obtained from different cell lines, virus strains and subtypes. Therefore, the status of mathematical models available for the description of key processes of the host cells themselves is still far from being satisfactory. However, despite all the complexity of eukaryotic host cells and the considerable diversity of viruses, it can be expected that extensive research efforts in this exciting field will finally allow us to derive fundamental laws that govern virus–host cell interaction and understand at least part of its enormous complexity.

3 Mathematical Models of Virus Dynamics

Mathematical modeling plays a crucial role for a quantitative understanding of intracellular virus dynamics. Based on the theoretical framework of such a model it is possible to systematically structure experimental data, to analyze biological systems of high complexity and to predict the system's behavior at any moment of time. Additionally, a model allows the investigation of parameter sensitivity and robustness of biological systems, e.g. to examine the influence of changes in intracellular metabolite concentrations, protein synthesis and degradation rates or attachment rates on virus yield per cell. Based on a model it is eventually possible to identify key components and their relationships as well as bottlenecks within biological systems, and to use this information for optimization and design of bioprocesses. However, no matter what approach is finally used one key question should be always answered beforehand: "what its use is and what problem it is intended to help to solve" [2].

Quantitative mathematical models of biological systems can be set up in several ways, ranging from mechanistic descriptions based on algebraic expressions and differential equations to stochastic approaches. A classification for cell populations, which has found general acceptance in biochemical engineering, was introduced by Frederickson [2,16,58]. In this context, the simplest models of virus dynamics can be formulated by *unstructured* or *structured* approaches [32]. An “unstructured” model does not consider intracellular phenomena. For example, the basic dynamics of viral growth in a bioreactor would be modeled by a system of three differential equations which describe how the number of uninfected cells, infected cells and free virus particles changes over time. Assuming that cells do not differ in composition, cell cycle phase, etc. (“average cell”), the cell population is designated “unsegregated”. In contrast, in a “structured” approach, different state variables are used to model virus replication in different cellular compartments such as the membrane, endosome, cytoplasm or nucleus. Based on genetic and molecular mechanistic data, rate equations are expressed for viral transcription, translation, protein expression and reactions catalyzed by virus-encoded enzymes. The most complex description is obtained by detailed models of intracellular virus growth and spreading in a population of heterogeneous individual cells [24]. Obviously, such a “structured segregated” approach represents the most realistic scenario to describe the progression of viral infections in populations of cells or individuals.

3.1 Unstructured Models of Virus Dynamics

Basic intracellular virus dynamics can be described by an approach developed by Rabinovitch and coworkers for the replication of bacteriophage T4 in *E. coli* [48]. The infection cycle is described by three parameters: an eclipse period ν until the first phage is completed in the bacterium, a constant rise rate α at which intracellular phages accumulate and a latent period λ (eclipse period + rise period) when the bacterium bursts and phages are released into the extracellular medium (Figure 2). The burst size B that corresponds to the number of released phages after cell lysis is defined by:

$$B = \alpha(\lambda - \nu).$$

Typical values for the burst size B are in the range of 10 to several hundred bacteriophages produced per infected bacterium after an eclipse period ν of 18–42 min at a rise rate α of 2–47 phages min^{-1} depending on experimental conditions and corresponding *E. coli* strain. Similar results were obtained by You and coworkers [64] who investigated the growth of phage T7 in *E. coli* in continuous culture under different dilution rates in a chemostat.

Another unstructured approach comprises the description of virus dynamics in a population of host cells by a system of differential equations. As an example, a basic model that describes the replication of influenza A virus in adherent Madin-Darby canine kidney (MDCK) cells in a bioreactor for inactivated vaccine production is briefly discussed in the following [39]. The time-delayed model considers three variables: the number of uninfected cells

U_C, infected cells I_C and free virus particles V. The number of uninfected cells U_C increases with a specific cell growth rate μ, while cells die with a specific cell death rate k_{cdf} due to process conditions. The rate of infection is proportional to the product of the concentration of uninfected cells U_C, the concentration of free virions V, and the virus infection rate k_{vi}:

$$\frac{\mathrm{d}U_\mathrm{C}}{\mathrm{d}t} = \mu \cdot U_\mathrm{C} - k_\mathrm{cdf} \cdot U_\mathrm{C} - k_\mathrm{vi} \cdot U_\mathrm{C} \cdot V \; .$$

Infected cells I_C disintegrate with a specific cell death rate k_{cdv}, mainly due to virus induced cell damage:

$$\frac{\mathrm{d}I_\mathrm{C}}{\mathrm{d}t} = k_\mathrm{vi} \cdot U_\mathrm{C} \cdot V - k_\mathrm{cdv} \cdot I_\mathrm{C} \; .$$

After a time delay τ (for m.o.i. > 1 corresponding to the latent period defined above) infected cells start to release progeny virus into the extracellular medium. The specific virus replication rate is μ_{vir}. Released virions are either degraded with a specific rate k_{vd} or adsorbed by uninfected, susceptible cells U_C with a specific attachment rate k_{va}:

$$\frac{\mathrm{d}V}{\mathrm{d}t} = \mu_\mathrm{vir} \cdot I_\mathrm{C}(t - \tau) - k_\mathrm{vd} \cdot V - k_\mathrm{va} \cdot U_\mathrm{C} \cdot V \; .$$

Adherent MDCK cells only proliferate in a bioreactor if attached to a microcarrier and growth slows down by contact inhibition. Main substrates of energy metabolism such as glucose and glutamine, as well as inhibiting metabolites such as lactate and ammonium, do not reach critical levels during virus replication [20]. Therefore, the degree of contact inhibition, determined by the maximum number of cells supported by the surface area of the carriers C_{max} (about 500) and the capacity used by infected and uninfected cell, is regarded as the only influence reducing the maximum growth rate μ_{max}:

$$\mu = \mu_\mathrm{max} \cdot \frac{C_\mathrm{max} - (U_\mathrm{C} + I_\mathrm{C})}{C_\mathrm{max}} \; .$$

Experimental results show that it takes about 4–6 h before visibly infected cells can be identified by light microscopy followed by a strong increase in virus titers [measured by a hemagglutination (HA) assay, an indirect method to quantify virus particles in suspension] after 10–12 h in the supernatant (Figure 3). After infection at low m.o.i. and typical process conditions, the maximum virus yield V_{max} is about 5×10^9–1×10^{10} virions ml^{-1} with a burst size of about 10 000–20 000 virus particles per infected cell. Simulation results show that small variations (below 10%) in initial values and specific rates do not influence V_{max} significantly. The main parameters relevant for obtaining maximal virus yields are specific virus replication rate and specific

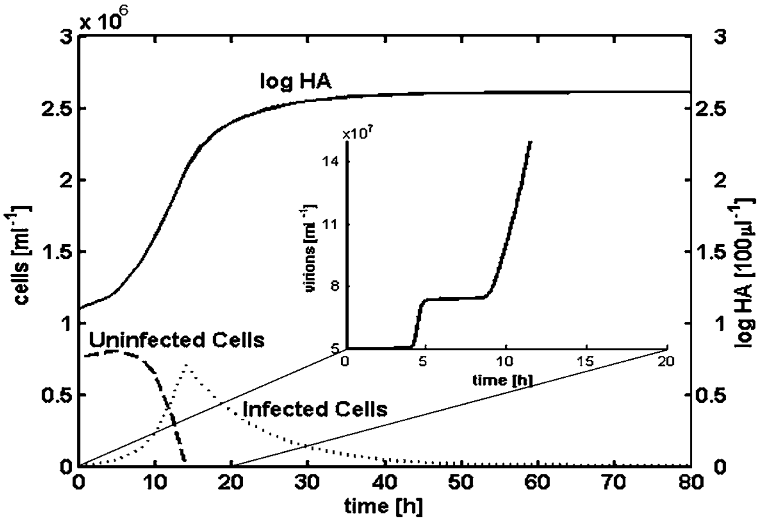


Figure 3 Unstructured time delay model for influenza A virus infection of a population of MDCK cells in a bioreactor at low m.o.i.: uninfected cells (dashed line), infected cells (dotted line) and the total number of virions produced (solid line) expressed in log HA units [39]. Inset: increase in virus particle number during the first hours post infection.

cell death rate due to infection. Similar models focusing on biotechnological applications of baculoviruses have been developed by Licari and Bailey [32], Nielsen [43], Enden and coworkers [12], and others. In addition, models with an equivalent structure have been introduced by Gilchrist and coworkers [21] and Nelson and coworkers [41] to describe immunological aspects and infectious diseases. An excellent overview on virus dynamics in micro-epidemiology is given by Nowak and May [44].

3.2 Structured Models of Virus Dynamics

Structured approaches have been developed to investigate one-step virus infection cycles in bacteria, insect and animal cells. A mathematical model for bacterial viruses, considered by Endy and coworkers [13], examines the replication of bacteriophage T7 in *E. coli*. The model provides a complete picture of the phage growth cycle. It describes individual steps of infection, including transcription of DNA, translation of mRNA and progeny phage assembly. In a similar way the quantitative intracellular kinetics of HIV type 1 was modeled by Reddy and Yin [50]. Typically, these models consist of a system of coupled differential equations comprising 10–50 differential equations and a minimum of 40–100 parameters and initial conditions. In addition, models that include stochastic events have been introduced (e.g. Ref.

[55]). As an example for structured models of virus replication, a dynamical model for the life cycle of influenza A virus in animal cells is discussed in more detail in the following Section.

4 Influenza Virus as an Example for Virus–Host Cell Interaction

Influenza virus is a lipid-enveloped negative-strand RNA virus that belongs to the Orthomyxoviridae family. Among the three types of influenza (A, B and C), influenza A virus is the best characterized. It causes respiratory infections that result in severe human and animal suffering with high economic losses. Conservative estimations indicate that in case of a human influenza pandemic 2–7 million people would die and tens of millions would need medical attention [61]. During the annual epidemics about 5–15% of the world population is affected with 250 000–500 000 deaths [60]. Influenza virus replication was thoroughly investigated by many groups, and a number of excellent books and reviews give a detailed description of its biological properties, the molecular composition of virions and the life cycle in its host cells [15,31,36,42,59].

Due to the detailed characterization of structure and life cycle of the influenza A virus, the availability of well-characterized cell systems for studying virus propagation in animal and human cell lines under controlled conditions and its relevance as a human pathogen, influenza A virus is (next to bacteriophages and HIV) an excellent “model virus” for systems biology. So far, the overwhelming number of studies on the influenza A virus life cycle have focused on the qualitative characterization of virus attachment, endocytosis, protein expression, genome replication, budding and release. In addition, a few mathematical models are available which describe molecular mechanisms of individual steps such as virus binding and endocytosis (e.g. Refs. [22, 38, 45]). In the following a first step towards a structured mathematical model, which describes the complete influenza A life cycle in animal cells is presented. The purpose of this model is 2-fold: (i) to analyze the key processes of virus replication at a cellular level, and (ii) to better understand rate-determining steps and factors limiting intracellular virus growth in mammalian cells for the design and optimization of virus-related production processes for recombinant proteins, classical vaccines and viral vectors for gene therapy. In the following sections key steps of the influenza A virus life cycle are described in more detail (Section 4.1). Then a structured mathematical model is introduced (Section 4.2) with simulation results for influenza virus growth dynamics (Section 4.3). The section ends with a discussion and an outlook on several aspects of influenza–host cell interaction not included into mathematical models so far (Section 4.4).

4.1 The Influenza A Virus Life Cycle

A schematic overview on the key steps of influenza A virus replication in animal cells, for example in adherent MDCK cell used for vaccine production [20], is given in Figure 4. The influenza A virus life cycle comprises the following steps:

1) Attachment to the apical membrane of the host cell and receptor-mediated endocytosis.
2) Uncoating and transfer of viral ribonucleoprotein complexes (vRNPs) into the nucleus.
3) Viral genome replication: transcription of (–) strand viral RNA [vRNA(–)] to (+) strand viral messenger RNA [vmRNA(+)].
4) Synthesis of complementary (+) strand RNAs [cRNAs(+)] and then vRNAs(–).
5) Translation of viral capsid, nonstrucctural and matrix proteins from vmRNAs(+).
6) Translation of viral envelope proteins from vmRNAs(+).
7) Packaging of progeny vRNPs and export to apical cell membrane.
8) Budding and release of progeny virions from the plasma membrane.

Influenza A virus infects humans and a wide variety of animals, including pigs, horses, ferrets and birds. In mammals, the virus replicates in epithelial cells of the upper respiratory tract, but *in vitro* it can also infect many other cell types that possess sialic acid-containing cellular surface proteins. The genome of influenza A virus is segmented and consists of eight vRNAs(–) of different size associated with nucleoproteins (NP) and three polymerases (PB1, PB2 and PA) involved in viral RNA transcription and viral genome replication. PB1 is a catalytic subunit of the viral RNA transcriptase and replication complex which catalyses the nucleotide addition during the elongation of the vmRNA(+) transcript. In addition, it has endonuclease activity. PB2 is a cap-dependent endonuclease, which is responsible for both binding and cleavage of capped cellular mRNA to generate primers for vmRNA(+) synthesis. PA is essential for viral genome replication; however, its actual role is not fully understood. The arrangement of vRNAs(–), NP proteins and the three associated polymerase proteins is called the vRNP or nucleocapsid. The vRNP complexes are enclosed within a shell composed of matrix protein (M1, the major component of virions) associated with a lipid membrane derived from the host cell. Embedded in the plasma membrane are three envelope proteins:

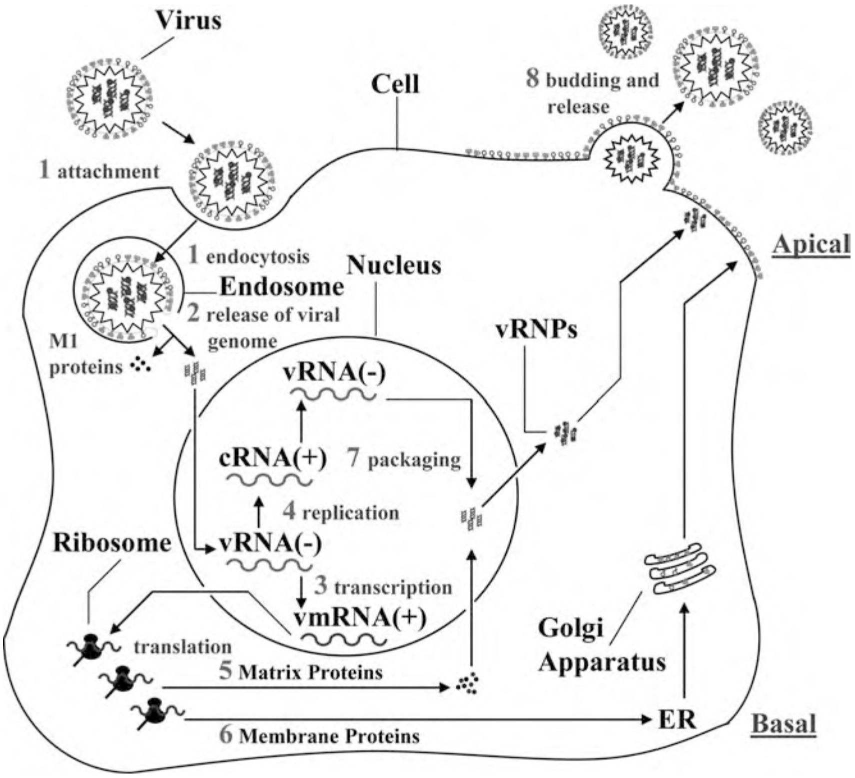


Figure 4 Single-cell reproductive cycle of influenza A virus. Different steps are numbered according to the virus life cycle discussed in Section 4 (adapted from Ref. [52] with permission).

HA, the major surface protein with sialic acid binding and membrane fusion activity at low pH, neuraminidase (NA) involved in virus release and an ion channel protein (M2). In addition, influenza virions contain NS2 proteins, also called *nuclear export proteins*.

Each of the viral genes has been sequenced completely for influenza A virus PR/8/34 and many other isolates of the different virus subtypes [19]. Eight genome segments encode 10 viral proteins. Nine of them are incorporated into the virus particle, except for the nonstructural protein NS1 – a regulatory factor with several activities which include the inhibition of cellular pre-mRNA 3′-end cleavage and polyadenylation, the inhibition of cellular pre-mRNA splicing in the nucleus, and the inhibition of the interferon-mediated cellular response to virus infection. Six genome segments encode one viral protein. The others contain two open reading frames (ORF), encoding for M1 and M2 proteins and NS1 and NS2 proteins, respectively. Recently a novel protein PB1-F2 was discovered, which is encoded by a second ORF, within the viral

RNA of PB1. The protein seems to participate in the induction of apoptosis and also seems to be involved in killing host immune cells responding to influenza virus infections. This protein was shown to be nonessential for virus replication *in vitro* and is therefore not considered in the model.

The following individual steps of the virus life cycle (Figure 4) including modeling assumptions concerning initial conditions, cellular resources and replication mechanisms are considered in the model.

(i) *Virus entry into the host cell.* Influenza virus particles attach to *N*-acetylneuraminic (sialic) acid-containing receptors on their host cell via the viral HA glycoprotein. In a next step, the virions are taken up into the cell by a clathrin-dependent receptor-mediated endocytosis. While several other entry pathways for influenza virus are also reported [53, 54], the model considers only the endocytic pathway, which seems to be the most common.

(ii) *vRNP uncoating and transport into the nucleus.* The delivery of the vRNPs into the cytoplasm of their host cell is a multistep process. After a pH drop in the endosome to approximately pH 5.0, viral M2 ion channels are activated, which allows protons to enter the interior of the virus particle. Upon acidification, HA molecules undergo conformational rearrangements mediating the fusion of the viral membrane with the endosomal membrane. Acidification also facilitates the dissociation of vRNPs from M1 matrix proteins. As a consequence, individual vRNPs are released into the cytoplasm of their host cells. The import of vRNPs into the nucleus is mediated by nuclear localization signals, carried by NP proteins. Since M1 proteins inhibit the import into the nucleus through the nuclear pore complexes (NPCs), their detachment from vRNPs plays a crucial role at this step. Virus particles, which do not fuse with the membrane, e.g. virions with defective M2 ion channels, are eventually degraded by lysosomes.

(iii) *Transcription [vmRNA(+) production].* Three types of viral RNAs are synthesized in the cellular nucleus: viral mRNAs of positive polarity [vmRNA(+)], viral genomic RNAs of negative polarity [vRNAs(–)] and complementary RNAs of positive polarity [cRNAs(+)]. Influenza virus vmRNAs(+) contain a cap structure at the 5′ end and a poly(A) tail at the 3′ end, which are derived from cellular precursor mRNAs. Their synthesis is catalyzed by the viral polymerase complex and comprises several steps. Splicing of M and NS vmRNAs(+) also occurs in the nucleus, and seems to be regulated by NS1 proteins. Newly synthesized vmRNAs(+) are exported from the nucleus to the ribosomes in the cytoplasm via nuclear pore complexes (NPCs) or degraded.

(iv) *Viral genome replication.* Viral genome replication involves the synthesis of full-length cRNAs(+), which serve as templates for vRNA(–) strands. Newly replicated vRNAs(–) are used for the production of further vmRNAs(+) and cRNAs(+) as well as for the assembly of vRNP complexes. While PB1 and PB2 proteins carry out transcription, genome replication requires PB1 and PA subunits of the viral polymerase complex. It was also shown that NP proteins not only mediate the nuclear import of vRNPs, but also participate in the switch from vmRNA(+) synthesis to viral genome replication. In addition it is assumed that all viral RNAs are degraded to a certain extent in the nucleus.

(v) *Capsid, nonstructural and matrix protein production.* Viral proteins (PB1, PB2, PA, NP, NS1, NS2 and M1) are synthesized at maximum rate by ribosomes organized in polysome complexes. At the same time the translation of cellular mRNAs seems to be inhibited. The mechanisms of a selective translation of vmRNAs(+) are poorly understood so far. There are three possible mechanisms for the inhibition of cellular protein synthesis [46]. One of them involves the degradation of cellular precursor mRNAs in the nucleus. Another option is the inhibition of the translation of cellular mRNAs at the initiation and elongation steps. Finally, cellular protein production seems to be inhibited by retarding the transport of cellular mRNAs to the cytoplasm. The newly synthesized polymerases as well as matrix and nonstructural proteins are transported to the nucleus, where they participate in M and NS mRNA splicing, transcription and viral genome replication. Additionally, they are consumed for the production of new vRNP complexes.

(vi) *Envelope protein production.* M2, HA and NA protein synthesis is carried out by membrane-bound ribosomes. Newly synthesized envelope proteins are translocated across the membrane of the endoplasmic reticulum (ER), glycosylated and transported out of the ER to the Golgi apparatus. Finally, they are delivered to the apical membrane of their host cell for the assembly with vRNP complexes and budding.

(vii) *Packaging.* The formation of vRNP complexes takes place in the cellular nucleus. Newly synthesized PB1, PB2, PA, NP and NS2 proteins associate with vRNAs and M1 proteins forming vRNP–M1–NS2 complexes, which mediate nuclear export. M1 proteins inhibit the re-import of vRNP complexes, while nuclear export signals of NS2 proteins seem to be required to overcome the nuclear import signals of NP proteins.

(viii) *Virus budding and release.* In the last step vRNP–M1–NS2 complexes interact with the cytoplasmic tails of M2, HA and NA proteins, which leads to the formation of a bud at the apical membrane of polarized epithelial cells. Host cell membrane proteins seem to be excluded from the newly formed progeny virions. Eventually, buds separate from the cellular surface and virions are released to the extracellular medium. The model assumes that eight influenza virus vRNP complexes are packaged per virion.

4.2 Mathematical Model of the Influenza A Virus Life Cycle

The model considers an "average" cell, which is immersed in a small volume of medium (1 nL) and infected by a low number of influenza virus particles (10 virions cell^{-1}), which survives about 10–12 h post-infection (p.i.). Only one replication cycle is taken into account without re-infection of the cell by newly released progeny virus. Additionally, it is assumed that cellular protein synthesis is shut off after infection and that infected cells do not divide. The model is represented by a system of 43 nonlinear ordinary differential equations (ODE) and 81 parameters. About half of these parameters (42) are confirmed by our own experiments or taken from literature. The non-confirmed parameters (39), e.g. degradation rate coefficients and parameters describing the switch from vmRNA(+) synthesis to viral genome replication, were estimated according to the average duration of individual replication steps and overall virus dynamics. In the following only aspects of the virus life cycle that are relevant for the results presented later (Section 4.3) are briefly discussed. A complete list of all equations, initial conditions, and kinetic parameters is given in the Appendix.

Infection starts with the virus entry into the host cells (also see Appendix, 1):

$$\frac{\mathrm{d}V_{\mathrm{ex}}}{\mathrm{d}t} = -k_{\text{ex-s},V_{\mathrm{ex}}} V_{\mathrm{ex}} + k_{\text{s-ex},V_{\mathrm{ex}}} V_{\mathrm{s}}$$

$$\frac{\mathrm{d}V_{\mathrm{s}}}{\mathrm{d}t} = k_{\text{ex-s},V_{\mathrm{s}}} V_{\mathrm{ex}} + k_{\text{s-ex},V_{\mathrm{s}}} V_{\mathrm{s}} - k_{\text{s-end}} V_{\mathrm{s}}$$

where t is the time measured in hours, V_{ex} and V_{s} represent the number of virus particles in the extracellular medium and on the cellular surface. The rate coefficient of virus binding to the cellular surface receptors is $k_{\text{ex-s},V_{\mathrm{ex}}}$. The rate constant of virus dissociation from the surface is $k_{\text{s-ex},V_{\mathrm{ex}}}$. The rates $k_{\text{ex-s},V_{\mathrm{s}}}$ and $k_{\text{s-ex},V_{\mathrm{s}}}$ are related to $k_{\text{ex-s},V_{\mathrm{ex}}}$ and $k_{\text{s-ex},V_{\mathrm{ex}}}$, respectively, via $k_{\text{s-ex},V_{\mathrm{s}}} V_{\mathrm{s}} = \frac{U_{\mathrm{r}}}{N_{\mathrm{cells}}} k_{\text{ex-s},V_{\mathrm{ex}}}$, and $k_{\text{s-ex},V_{\mathrm{s}}} = \frac{U_{\mathrm{r}}}{N_{\mathrm{cells}}} k_{\text{s-ex},V_{\mathrm{ex}}}$, where U_{r} is the medium volume, containing N_{cells} cells ($N_{\mathrm{cells}} = 1$). Usually, cells are infected with a small number of virions. Therefore, the number of cellular receptors, which is in the range of 10^4–10^5 receptors per cell (see Appendix, Table A1), significantly exceeds the number of virus particles binding to the cell surface and does not represent a limiting factor for virus entry. Consequently, a Michaelis–Menten-type kinetics is not considered and it is assumed that $k_{\text{ex-s},V_{\mathrm{ex}}}$ as well as $k_{\text{s-ex},V_{\mathrm{ex}}}$ is constant. Virus uptake via clathrin-coated pits is omitted in the model and viruses attached to the cellular membrane are assumed to penetrate directly into endosomes. Typically, there are several hundred endosomes per cell. Therefore, their number significantly exceeds the number of endosomes containing virus and the rate of endocytosis is assumed constant.

The next steps comprise vRNP uncoating and transport into the nucleus (see Appendix, 2) via NPCs, which is in the range of 3000–4000 pores per cell. At low m.o.i. only a small percentage of the NPCs is involved in the transfer of vRNPs into the nucleus at time of infection. With the transfer of vRNPs into the nucleus the synthesis of vmRNAs(+) starts (also see Appendix, 3):

$$\frac{\mathrm{d}C_{i,\mathrm{nuc}}}{\mathrm{d}t} = k_{\mathrm{v\text{-}vm},i} P_{\mathrm{Pol,nuc}} - k_{\mathrm{vm},i,\mathrm{nuc\text{-}cyt}} C_{i,\mathrm{nuc}} - k_{\mathrm{vm},i,\mathrm{nuc\text{-}degr}} C_{i,\mathrm{nuc}}$$

$$\frac{\mathrm{d}C_{i,\mathrm{cyt}}}{\mathrm{d}t} = k_{\mathrm{vm},i,\mathrm{nuc\text{-}cyt}} C_{i,\mathrm{nuc}} - k_{\mathrm{vm},i,\mathrm{cyt\text{-}degr}} C_{i,\mathrm{cyt}}$$

$$i \in [\mathrm{Pol, NP, M1, NS1, NS2, M2, HA, NA}]\,,$$

where $C_{i,\mathrm{nuc}}$ is the number of vmRNAs(+) encoding the i-th protein in the nucleus, $C_{i,\mathrm{cyt}}$ is the number of vmRNAs(+) in the cytoplasm, and $P_{\mathrm{Pol,nuc}}$ is the number of polymerase complexes in the nucleus. The first term ($C_{i,\mathrm{nuc}}$) in the right-hand side of the first equation is the rate of vmRNA(+) synthesis in the nucleus. All three polymerase subunits and their corresponding vRNA(–) are considered as one unit. The polymerase complexes are assumed to operate at the same speed, so the first term is proportional to $P_{\mathrm{Pol,nuc}}$. The rate coefficient $k_{\mathrm{v\text{-}vm},i}$ of vmRNAs(+) synthesis depends on the number of vRNAs(–) in the nucleus C_{v} (see Appendix, 4.2) and on $P_{\mathrm{NP,nuc}}$, the number of NP molecules in the nucleus which are assumed to regulate the switch from vRNA(–) transcription to viral genome replication depending on their nuclear concentration (see Appendix, 5.2). In addition, it is assumed that the rate of the process is not limited by C_{v}. Every polymerase complex is involved in transcription and all polymerase complexes operate at the same speed. As mentioned above, NP proteins are assumed to inhibit the production of vmRNAs(+). Consequently, $k_{\mathrm{v\text{-}vm},i}$ should be maximal when $P_{\mathrm{NP,nuc}} = 0$ and should tend to zero when $P_{\mathrm{NP,nuc}} \to \infty$. Taking all this into account, it is assumed that $k_{\mathrm{v\text{-}vm},i} = k_{\mathrm{v\text{-}vm},i,\mathrm{max}} \frac{1}{1+a_{\mathrm{NP}} P_{\mathrm{NP,nuc}}}$, where a_{NP} is a positive parameter. Here, a_{NP} represents an inverse concentration of NP proteins at which $k_{\mathrm{v\text{-}vm},i} = \frac{1}{2} k_{\mathrm{v\text{-}vm},i,\mathrm{max}}$. It is a_{NP} that defines the influence of NP proteins on vmRNA(+) production. The rate coefficients of viral mRNA nuclear export and degradation are $k_{\mathrm{vm},i,\mathrm{nuc\text{-}cyt}}$ and $k_{\mathrm{vm},i,\mathrm{nuc\text{-}degr}}$, respectively. The rate constant of cytoplasmic vmRNA(+) degradation is $k_{\mathrm{vm},i,\mathrm{cyt\text{-}degr}}$. A splicing of M and NS mRNAs is not considered in the model.

Viral genome replication is modeled in a similar way (see Appendix, 4):

$$\frac{\mathrm{d}C_{\mathrm{c}}}{\mathrm{d}t} = k_{\mathrm{v\text{-}c}} P_{\mathrm{Pol,nuc}} - k_{\mathrm{c\text{-}degr}} C_{\mathrm{c}}$$

$$\frac{\mathrm{d}C_{\mathrm{v}}}{\mathrm{d}t} = k_{\mathrm{c\text{-}v}} P_{\mathrm{Pol,nuc}} + k_{\mathrm{spl},C_{\mathrm{v}}} S_{\mathrm{nuc}} - k_{\mathrm{un},C_{\mathrm{v}}} C_{\mathrm{v}} \prod_{l} P_{l,\mathrm{nuc}} - k_{\mathrm{v\text{-}degr}} C_{\mathrm{v}}\,,$$

$l \in [\mathrm{Pol}, \mathrm{NP}, \mathrm{M}, \mathrm{NS2}]$,

where C_c is the number of cRNAs(+), C_v is the number of vRNAs(–) and $P_{l,\mathrm{nuc}}$ is the number of molecules of the l-th protein in the nucleus. The rate coefficients of cRNA(+) and vRNA(–) synthesis are $k_{\text{v-c}}$ and $k_{\text{c-v}}$, respectively. Assuming that all vRNA molecules are synthesized at similar rates [63], it is not necessary to describe their numbers by different functions. Furthermore, it is assumed that every polymerase complex participates in either positive or negative strand synthesis. Neither C_v nor C_c limit the rate of the corresponding process because their number exceeds the number of polymerase complexes. Since a high concentration of NP proteins activates genome replication, $k_{\text{v-c}}$ and $k_{\text{c-v}}$ should have maximum values when $P_{\mathrm{NP,nuc}} \to \infty$ and be equal to zero when $P_{\mathrm{NP,nuc}} = 0$. Thus, it is assumed that $k_{\text{v-c}} = k_{\text{v-c,max}} \frac{P_{\mathrm{NP,nuc}}}{b_{\mathrm{NP}} + P_{\mathrm{NP,nuc}}}$ and $k_{\text{c-v}} = k_{\text{c-v,max}} \frac{P_{\mathrm{NP,nuc}}}{b_{\mathrm{NP}} + P_{\mathrm{NP,nuc}}}$. Here, b_{NP} is a positive parameter that defines the influence of NP proteins on viral genome replication. It represents the concentration of NP proteins at which $k_{\text{v-c}} = \frac{1}{2} k_{\text{v-c,max}}$ and $k_{\text{c-v}} = \frac{1}{2} k_{\text{c-v,max}}$. The rate constants of cRNA(+) and vRNA(–) degradation are $k_{\text{c-degr}}$ and $k_{\text{v-degr}}$, respectively. The rate constants of vRNP splicing $k_{\mathrm{spl},C_\mathrm{v}}$ and $k_{\mathrm{spl},S_\mathrm{nuc}}$ (see Appendix, 2.2) are given by $k_{\mathrm{spl},C_\mathrm{v}} = C_{\mathrm{seg}} k_{\mathrm{spl},S_\mathrm{nuc}}$ and $k_{\mathrm{spl},P_{i,\mathrm{nuc}}} = P_{i,\mathrm{seg}} k_{\mathrm{spl},S_\mathrm{nuc}}$, respectively, where C_{seg} is the average number of nucleotides contained in one segment. [Influenza A (A/PR/8/34) consists of about $C_{\mathrm{vir}} = 13588$ nucleotides [31]; therefore one segment has an average of $C_{\mathrm{seg}} = C_{\mathrm{vir}} / N_{\mathrm{seg}} = 1699$ nucleotides.] The rate constant of the assembly of new vRNPs is $k_{\mathrm{un},C_\mathrm{v}}$.

Translation of viral proteins includes the synthesis of capsid, nonstructural and matrix proteins (polymerase, NP, M1, NS1 and NS2; see Appendix, 5), which are transported from the cytoplasm back into the nucleus. Newly synthesized envelope proteins (HA, NA, M2; see Appendix, 6) are further processed via the ER and Golgi apparatus. The influence of the number of cellular ribosomes on viral protein synthesis is neglected, assuming that a cell contains about $R_0 = 5 \times 10^6$ ribosomes, which is significantly higher than the number of vmRNA(+) molecules to be processed. In addition, it is assumed that the ribosomes are organized in polysome complexes with an average distance of about 80 nucleotides. Furthermore, it is assumed that rate coefficients of the synthesis of viral proteins do not depend on the cellular pool of free amino acids P_{cell} for the following reasons: (i) consumption of amino acids for cellular protein synthesis ceases soon after infection [46]; (ii) uninfected cells contain a pool of approximately $P_{\mathrm{cell}} = 3.1 \times 10^{10}$ free amino acids (0.4% of cellular wet weight, 138 Dalton average weight of amino acids; [1, 40], which is sufficient to produce about 1.3×10^4 virions (about 2.4×10^6 amino acids per virion, influenza A/PR/8/34) even when assuming that after infection cellular energy and amino acid metabolism is not switched off and there could be an additional supply of amino acids for viral protein synthesis [20]. Thus, the influence of virus infection on the pool of cellular amino acids and therefore on the rate of viral protein synthesis seems to be negligible.

While envelope proteins are incorporated into the apical membrane of the cell vRNPs associate in the nucleus and are being transported to the budding site (see Appendix, 7). It is assumed that vRNPs are packaged randomly with eight segments each. Finally, the assembly of virions is completed and virus particles are released into the medium (see Appendix, 8):

$$\frac{\mathrm{d}V_{\mathrm{bud}}}{\mathrm{d}t} = k_{\mathrm{bud},V_{\mathrm{bud}}} S_{\mathrm{un,bud}} \prod_{l} P_{l,\mathrm{bud}} - k_{\text{bud-rel},V_{\mathrm{bud}}} V_{\mathrm{bud}}$$

$$l \in [\mathrm{M2}, \mathrm{HA}, \mathrm{NA}]$$

$$\frac{\mathrm{d}V_{\mathrm{rel}}}{\mathrm{d}t} = k_{\text{bud-rel},V_{\mathrm{rel}}} V_{\mathrm{bud}} \ ,$$

where V_{bud} and V_{rel} are the number of budding and released virions, respectively. The rate constant of the assembly of progeny virus particles $k_{\mathrm{bud},V_{\mathrm{bud}}}$ is related to $k_{\mathrm{bud},P_{j,\mathrm{bud}}}$ and $k_{\mathrm{bud},S_{\mathrm{un,bud}}}$ via $k_{\mathrm{bud},P_{j,\mathrm{bud}}} = P_{j,\mathrm{vir}} k_{\mathrm{bud},V_{\mathrm{bud}}}$ and $k_{\mathrm{bud},S_{un,bud}} = N_{\mathrm{seg}} k_{\mathrm{bud},V_{\mathrm{bud}}}$ (see Appendix, 6.2 and 7.2). Here $P_{j,\mathrm{vir}}$ is the number of molecules of the j-th envelope protein in the virus particle and $k_{\text{bud-rel}}, V_{\mathrm{bud}}$ is the rate constant of progeny virus release. The rates $k_{\text{bud-rel},V_{\mathrm{rel}}}$ and $k_{\text{bud-rel},V_{\mathrm{bud}}}$ are related via $k_{\text{bud-rel},V_{\mathrm{bud}}} = \frac{U_{\mathrm{r}}}{N_{\mathrm{cells}}} k_{\text{bud-rel},V_{\mathrm{rel}}}$.

4.3 Influenza A Virus Growth Dynamics

Simulation results of the mathematical model discussed above for the influenza A virus dynamics are shown in Figures 5–10. Immediately after the infection the number of extracellular virions decreases and within about 2 h more than 90% are taken up by the cell (Figure 5). Correspondingly, both the number of virions attached to the surface and, with a short delay, the number of virions incorporated into endosomes increases and a maximum is achieved for virions attached to the surface and endosomal virions after 14 and 19 min, respectively (Figure 5). From the endosomes individual vRNPs are released into the cytoplasm and start to accumulate in the nucleus (Figure 6). The maximum number of parental vRNPs in the cytoplasm and in the nucleus is found 38 and 99 min p.i., respectively.

After about 25–27 min p.i. first vmRNAs(+) are synthesized in the nucleus and between 1 and 2 h p.i. the number of vmRNAs(+) in the nucleus increases significantly before it achieves a steady state due to continuous export into the cytoplasm and degradation (Figure 7). About 40–44 min p.i., first vmRNAs(+) are found in the cytoplasm, where they accumulate. The switch from vmRNA(+) production to viral genome replication starts about 55 min p.i. when

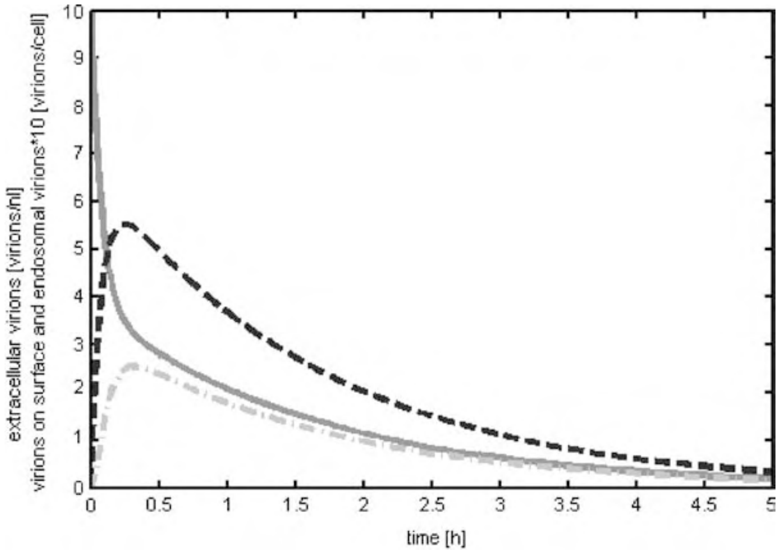


Figure 5 Virus attachment and endocytosis: free extracellular virions (solid line), virions attached to the cellular surface membrane (dashed line) and virions (dash/dotted line) incorporated by endocytosis.

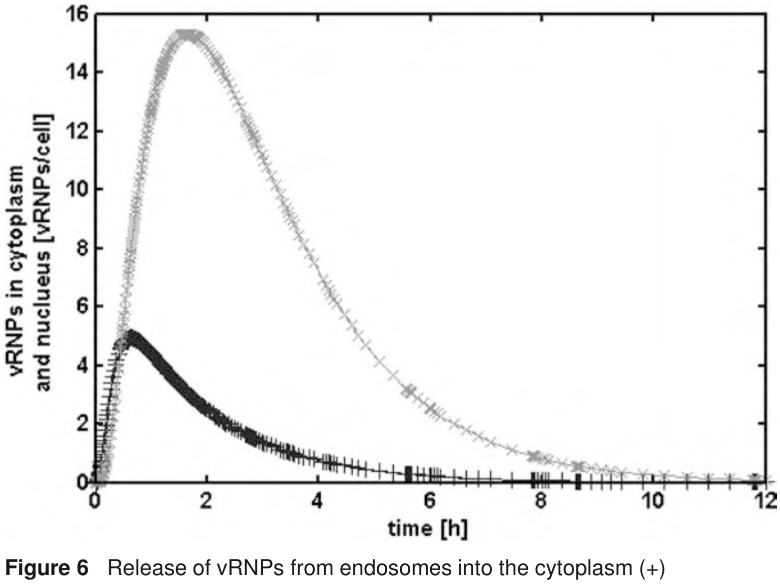


Figure 6 Release of vRNPs from endosomes into the cytoplasm (+) and accumulation of individual vRNPs in the nucleus (×).

a significant amount of newly produced NP proteins have accumulated in the nucleus (Figure 8). About 4 h p.i. cRNA(+) is synthesized at almost maximum rate. As a consequence, viral genome is replicated and the assembly of new vRNP complexes starts in the nucleus (Figure 9).

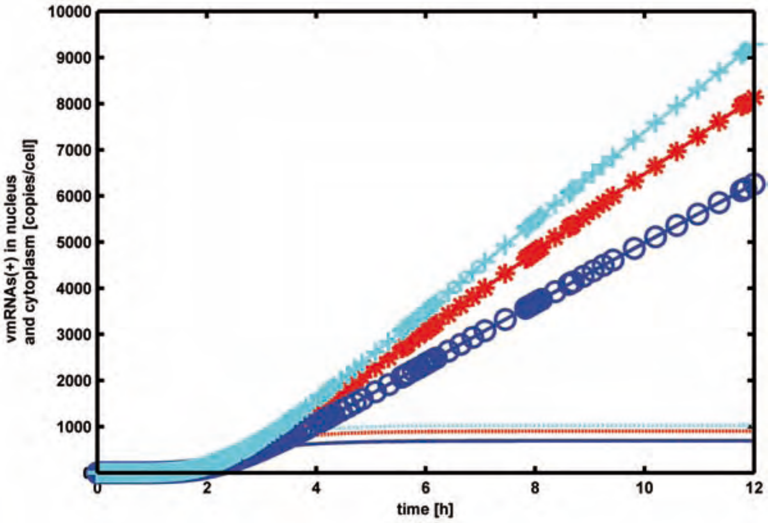


Figure 7 Transcription of viral genes. Accumulation of vmRNAs (+) in nucleus [NP (-· — ·-), HA (- - -), polymerase (—)] and cytoplasm [NP (+), HA (×), polymerase (○)].

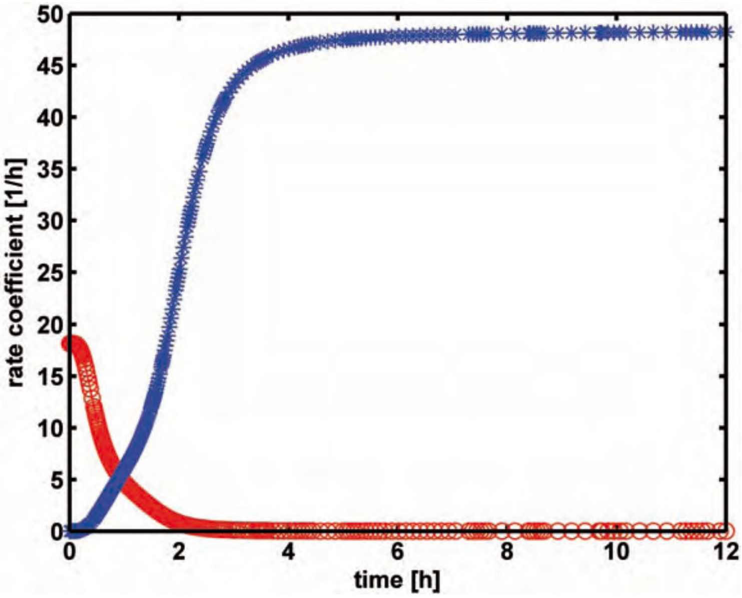


Figure 8 Switch from viral transcription to viral genome replication: rate coefficients of vmRNA (+) synthesis (○) and cRNA (+) production (×).

Approximately 2.5 h p.i., first virions are released into the extracellular medium. Initially, the number of released virions increases exponentially

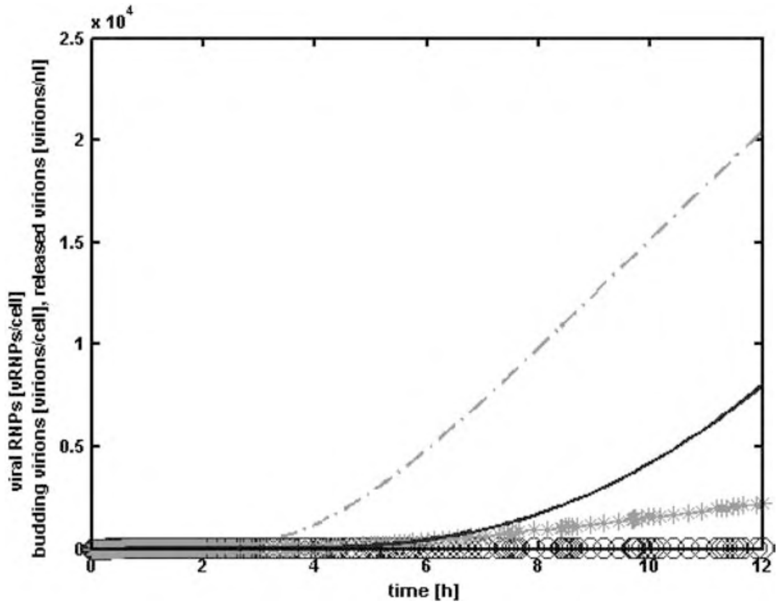


Figure 9 Dynamics of vRNPs: In the nucleus, vRNPs (–·–·–) accumulate while virion formation is limited by the number of vRNPs in the cytoplasm (-o-o-) during budding. Progeny virions are formed at the surface membrane of the host cell (–x–x–), and released into the extracellular medium (—). The maximum number of released virions is about 8000 per cell.

as vRNP formation and budding are not limited in precursor supply. As soon as viral proteins or vRNAs(–) become limiting, viral growth increases proportional to the square of time. [This can be explained by the fact that at the late period of infection, vmRNAs(+) are produced at maximum rate in the nucleus, and, as a result, the total number of all viral proteins and RNAs, as well as the number of budding viruses, increase linearly with time.] In total the cell produces about 8000 virions within 12 h (Figure 9) before it dies due to the virus interfering with basic cellular processes or virus induced apoptosis.

During the replication cycle several cellular resources [surface receptors, endosomes, transport capacity for vRNPs, vmRNAs(+) and viral proteins via the NPCs, precursors for viral protein and genome synthesis, ATP and redox equivalents in form of NAD(P)H, cellular plasma membrane] are required to a varying extent. During the early steps of virus replication neither the number of cellular surface receptors nor the number of endosomes is a limiting factor. Both resources exceed by far the capacities for virus internalization at low m.o.i. Virus attachment and detachment kinetics itself also does not seem to influence overall virus dynamics on the cellular level. Even drastic changes

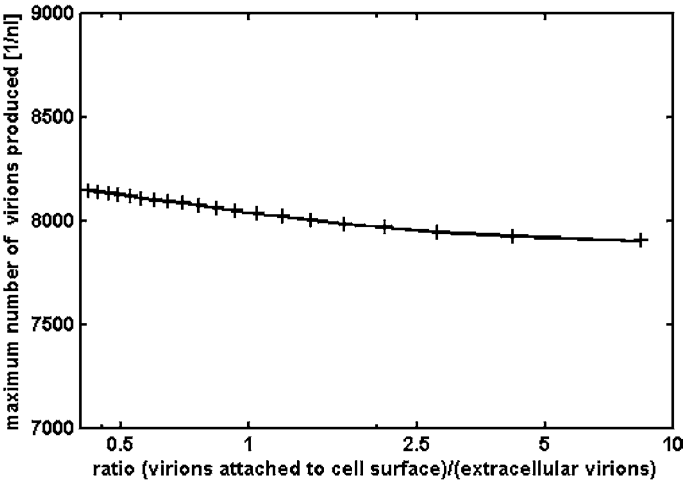


Figure 10 Influence of binding kinetics. Variations of the ratio (virions attached to the surface)/(extracellular virions) at steady-state have almost no influence on the maximum number of virus particles released 12 h p.i. (+).

in attachment kinetics have almost no impact on the total number of virions released during virus replication (Figure 10).

Transfer of vRNPs via the NPCs, which is in the range of 3000–4000 nuclear pores in mammalian cells, also does not seem to limit virus replication. Assuming a transfer rate of 10^2–10^4 molecules s^{-1}, typical for membrane transport proteins [34], 10^9–10^{11} viral molecules [vmRNAs(+), vRNPs, viral proteins] could cross the nuclear membrane per hour. Simulation results clearly demonstrate that NPCs are mainly used for the transfer of newly synthesized viral proteins, e.g. M1 and NP, from cytoplasm into the nucleus. The maximum rate of this process is about 10^7 molecules h^{-1} and therefore negligible with respect to total transfer capacity of the cell. The bottleneck during early steps of infection seems to depend on the virus itself. For a successful infection the fusion of the viral membrane with the endosomal membrane and the release of the viral genome into the cytoplasm of the cell is essential. According to Martin and Helenius [37], the genome of only 65–70% of all endosomal viruses is finally released into the cytoplasm. Additionally, packaging of the correct number and distribution of vRNPs is required to successfully infect a cell. Both factors together explain the comparatively low number of successful infections in influenza A virus infectivity studies. Depending on the cell system, cultivation conditions and assay (plaque forming units, tissue culture infectivity dose), the ratio of infectious to noninfectious virus particles varies in the range of 1/50–1/20 [15].

During the first hour of virus replication the total number of free cellular nucleotides and amino acids consumed for virus production makes up only a small part of the total cellular pool of these components. Even under the very restricting assumption that the cellular biosynthesis of nucleotides and amino acids as well as any uptake of these precursors from the medium is stopped early during infection, their total number is much higher than necessary for synthesizing about 4000–8000 virus particles within the average life-time of an infected cell which is in the range of 10–12 h. Assuming a cellular pool size of 1.3×10^{10} nucleotides and 3.1×10^{10} amino acids (Appendix, Table A1) at most 0.8% of the available nucleotides and 60.8% of the free amino acids are consumed for virion production.

The maximum number of vmRNAs(+) in the cytoplasm never exceeds 10^4 molecules cell^{-1} (not shown). Therefore, the number of ribosomes involved in the synthesis of viral proteins is significantly smaller than the total number of cellular ribosomes (approximately 10^7). Consequently, the number of cellular ribosomes does not seem to limit viral protein production. Also, the number of heterogeneous nuclear precursor mRNAs, which is about 2.2×10^5 molecules cell^{-1} [26], is not limiting for vmRNA(+) synthesis (not shown).

Simulations show that during vRNP assembly, as well as during virus budding, one of the newly produced viral components is completely consumed for the virion formation. Under the current modeling assumptions it is the number of M1 proteins that represents a limiting factor for the formation of new vRNP–M1–NS2 complexes in the nucleus while the number of all other viral proteins increases linearly with time (not shown). A similar situation takes place during virus budding where newly synthesized vRNPs represent the limiting factor (Figure 9). In contrast, viral membrane proteins seem to accumulate at the budding sites.

What factors mainly limit virus production? Simulation studies show that the total number of virus particles produced during the life-time of an infected cell mainly depends on the efficiency of viral polymerase complexes required for vmRNA(+) and therefore for viral protein synthesis. Even drastic variations in the parameters of all other steps during virus attachment/detachment, endocytosis, transfer of vRNPs into the nucleus result in only modest changes in infection dynamics, e.g. a time delay before vmRNA(+) synthesis starts or an initial reduction in the number of vRNPs entering the nucleus. However, as soon as the infection is successfully initiated the total number of virions produced during a fixed period of time is more or less constant. Changes in the ratio (virions attached to the surface)/(extracellular virions) at steady-state in the range of 0.4–8.4 (standard 2.8), for instance, result in differences of less than 2.5% in the total number of virus particles produced (Figure 10). In addition, the protein synthesis capacity of the infected cell itself has a major influence. Even when the number of ribosomes and the number of precursors itself do not seem to be limiting factors for viral protein synthesis the translation efficiency at the ribosomes clearly influences overall virus production. For example, an increase of the corresponding rate coefficient by a factor of 2 (k_{Rib}, see Appendix, Table A2) increases the total number of virus particles produced by a factor of 2.15. Overall, it seems that the capacity for influenza virus production in eukaryotic cells is mainly determined by the efficiency of protein synthesis at the ribosomes and the synthesis rate of vmRNAs(+) defined by the viral polymerase complexes. Cellular resources such as the number of ribosomes, the number of nucleotides and the number of available amino acids do not seem to limit virion formation during the average life-time of infected cells.

4.4 Discussion and Outlook

The structured model for the complete single-cell reproductive cycle of influenza A virus in mammalian cells presented takes into account most steps from virus entry into the host cell to progeny virus release. The overall virus dynamics agrees well with results obtained for other structured models, for example HIV-1 infection of lymphocytes [50] or the replication of bacteriophage T7 in *E. coli* [13].

However, even with the time-course of virus entry, vmRNA(+) synthesis, viral genome replication, virus budding and release reasonably represent overall virus dynamics on a cellular level there are several aspects of the virus replication cycle which should be quantitatively characterized in more detail. First of all, the control of vmRNA(+) synthesis is not completely understood. According to Lamb and Choppin [30], similar amounts of vmRNAs(+) and corresponding viral proteins are produced during the very first hours of infection only. Later in infection vmRNAs(+) coding for NP and NS1 proteins seem to be synthesized in higher amounts than those required for polymerases. After the onset of vRNA(–) production and vRNP association, the synthesis of vmRNAs(+) seems to dominate. Therefore, the assumption of the present model that all vmRNAs(+) as well as all viral proteins are produced in equal amounts is oversimplified. In addition, the vRNAs(–) seem to be produced in nonequivalent amounts. As mechanisms regulating a selective synthesis of viral proteins and genome segments are not understood quantitatively, no attempts have been made so far to introduce corresponding control factors into the existing model. A similar problem is the discussed switch from viral protein synthesis to viral genome replication. It seems that the number of NP proteins in the nucleus is regulating this switch, but the exact mechanism is not clear. Correspondingly, the parameters controlling the switch from vmRNA(+) production to cRNA(+) synthesis (see Appendix, 3 and 4) are more or less arbitrarily selected to follow the overall time course of virus replication. Clearly, more experiments are needed to better understand these complex phenomena. Finally, viral genome packaging itself is still a subject of discussion. So far, it is not clear whether genome segments are selectively incorporated into a progeny virus particle or packaging of genome segments is a purely random process. Experimentally, a ratio of infectious to noninfectious virus particles was confirmed to be in the range of 1/50–1/20 favoring selective mechanisms [15,17]. On the other hand, assuming that virus particles can contain more than eight genome segments [11, 42] results in similar ratios (1/90–1/50 for nine segments). So far, for simplicity it is assumed in the model that all virus particles contain eight genome segments. However, if required this can be easily modified to cope with corresponding experimental results.

Cells infected with influenza virus die at about 10–12 h p.i. So far, the mechanisms of cell death are not fully understood. As shown before, cellular resources consumed for virus replication are used only to a limited degree for virion formation. Therefore, cell death is probably not directly linked with the exhaustion of cellular pools. Also the number of nuclear precursor mRNAs [their poly(A) tail] used for vmRNA(+) production does not to seem to be a limiting factor (Section 4.3). The most likely reason for cell death is virus-induced apoptosis, an active process of cellular self-destruction. For influenza A virus several factors have been identified which activate apoptosis pathways [35] including caspase activation by double-strand vRNAs of infected cells [33, 57, 66], activation of the latent transforming growth factor-β by viral membrane proteins HA and NA [51] and the interaction with mitochondrial-dependent apoptotic pathways of a newly described influenza A virus gene product PB1-F2 [6]. In addition, host defense mechanisms such as interferon-mediated antiviral responses and virus induced cellular gene expression play important roles in virus replication [18,29,35,36]. However, while these topics are being studied in an increasing number of experimental systems, their kinetics as well as the precise mechanisms of apoptotic factors and virus-induced stimulation of cellular defense on virus replication still have to be elucidated in more detail before incorporation into mathematical models of virus replication.

5 Conclusions

Mathematical models of virus replication in single cells are a crucial step towards a quantitative understanding of virus–host cell interaction. Most models developed so far focus on kinetics of virus attachment and entry, transcription and translation of viral mRNAs, genome replication, assembly, and virus release. In particular, the overall growth dynamics of bacteriophages, the influence of the physiological status of the host bacterium and the interaction of phage components with the regulatory networks of their host cells have been analyzed in detail. Models of viruses infecting eukaryotic cells, especially mammalian cells, are adding a further level of complexity. Apart from the impact of virus replication on cellular metabolism, the impact of viral components on host cell defenses as well as on cellular signal transduction networks and gene expression have to be considered to fully understand the multiple facets of virus–host cell interaction. However, existing mathematical models developed for the simulation of HIV, influenza A virus and baculovirus dynamics neglect most of these aspects which differ eukaryotes from prokaryotes. Therefore, these models can be considered only as a first step towards a quantitative and integrative understanding of virus replication on a cellular level. For further progress, virus–eukaryotic cell systems should be selected as models for systems biology approaches to cope with these challenges and to improve our understanding on virus–host cell interactions. Eventually, this will allow us not only to reveal basic laws that control virus replication in cells and to identify molecular targets for virus-related diseases, but also to optimize viral-based production systems for vaccines and pharmaceuticals required for gene or cancer therapy.

Acknowledgments

The authors wish to thank Dietrich Flockerzi, Yvonne Genzel and Lars Möhler for thoughtful comments and discussions on virus–host cell interaction, and for providing experimental data as well as modeling and simulation results for influenza A virus replication.

Nomenclature

a_{NP}	influence of the NP protein on vmRNA production (cell NP^{-1})
b_{NP}	influence of the NP protein on genome replication (NP cell^{-1})
B	burst size (phages bacterium^{-1})
$C_{\max}$	maximum cell concentration due to contact inhibition (cells mL^{-1})
C_{c}	cRNA (nucleotides cell^{-1})
C_{cell}	number of free cellular nucleotides (nucleotides cell^{-1})
$C_{\mathrm{m,cell}}$	number of cellular mRNAs (nucleotides cell^{-1})

$C_{i,\mathrm{nuc}}$	vmRNA, encoding the *i*-th protein in the nucleus (nucleotides cell^{-1})
$C_{i,\mathrm{cyt}}$	vmRNA, encoding *i*-th protein in the cytoplasm (nucleotides cell^{-1})
C_{seg}	the average number of nucleotides of one vRNA segment (nucleotides)
C_{v}	number of vRNA nucleotides (nucleotides cell^{-1})
C_{vir}	number of nucleotides in a virus particle (nucleotides)
d_{rib}	distance between ribosomes processing a vmRNA (nucleotides)
E	number of cellular endosomes (endosomes cell^{-1})
H	number of nuclear pores (pores cell^{-1})
I_{C}	concentration of infected cells (cells mL^{-1})
$k_{\mathrm{bud\text{-}rel},V_{\mathrm{bud}}}$	rate constant of progeny virus release (h^{-1})
$k_{\mathrm{bud\text{-}rel},V_{\mathrm{rel}}}$	rate constant of progeny virus release (nL^{-1} h^{-1})
$k_{\mathrm{bud},P_{j,\mathrm{bud}}}$	rate constant of progeny virus particles assembly (h^{-1})
$k_{\mathrm{bud},S_{\mathrm{un,bud}}}$	rate constant of progeny virus particles assembly (h^{-1})
$k_{\mathrm{bud},V_{\mathrm{bud}}}$	rate constant of progeny virus particles assembly (h^{-1})
k_{cdf}	specific cell death rate due to cultivation conditions (h^{-1})
$k_{\mathrm{c\text{-}degr}}$	rate constant of cRNA degradation (h^{-1})
k_{cdv}	specific cell death rate due to viral infection (h^{-1})
$k_{\mathrm{c\text{-}v}}$	rate coefficient of vRNA synthesis (h^{-1})
$k_{\mathrm{cyt\text{-}nuc}}$	rate coefficient of vRNP nuclear import (h^{-1})
$k_{\mathrm{end\text{-}cyt},S_{\mathrm{cyt}}}$	rate coefficient of endocytosis (h^{-1})
$k_{\mathrm{end\text{-}cyt},V_{\mathrm{end}}}$	rate constant of vRNP uncoating (h^{-1})
$k_{\mathrm{end\text{-}degr}}$	rate constant of virus degradation within endosomes (h^{-1})
$k_{\mathrm{ex\text{-}s},V_{\mathrm{ex}}}$	rate coefficient of virus binding (nL h^{-1})
$k_{\mathrm{ex\text{-}s},V_{\mathrm{s}}}$	rate coefficient of virus binding (h^{-1})
$k_{i,\mathrm{cyt\text{-}degr}}$	rate constant of the degradation of the *i*-th protein in cytoplasm (h^{-1})

$k_{i,\text{cyt-nuc}}$ rate coefficient of nuclear import of the *i*-th protein (h^{-1})

$k_{i,\text{nuc-degr}}$ rate constant of the degradation of the *i*-th protein in the nucleus (h^{-1})

$k_{i,\text{synt}}$ rate coefficient of the synthesis of the *i*-th protein (h^{-1})

$k_{j,\text{bud-degr}}$ rate constant of the degradation of the *j*-th protein at the budding site (h^{-1})

$k_{j,\text{ER-bud}}$ rate constant of the transport of the *j*-th protein to the budding site (h^{-1})

$k_{j,\text{ER-degr}}$ rate constant of the degradation of the *j*-th protein in the ER (h^{-1})

k_{Pl} rate of the synthesis of RNA (nucleotides h^{-1})

k_{Rib} rate of the elongation of a peptide chain (amino acids h^{-1})

$k_{\text{s-end}}$ rate coefficient of the endocytosis (h^{-1})

$k_{\text{s-ex},V_{\text{ex}}}$ rate constant of virus dissociation from the surface (h^{-1})

$k_{\text{s-ex},V_{\text{s}}}$ rate constant of virus dissociation from the surface (h^{-1})

$k_{\text{spl},C_{\text{v}}}$ rate constant of vRNP splicing (h^{-1})

$k_{\text{spl},P_{i,\text{nuc}}}$ rate constant of vRNP splicing (h^{-1})

$k_{\text{spl},S_{\text{nuc}}}$ rate constant of vRNP splicing (h^{-1})

$k_{\text{un},C_{\text{v}}}$ rate constant of the assembly of new vRNP (h^{-1})

$k_{\text{un,nuc-bud}}$ rate coefficient of the export of new vRNP (h^{-1})

$k_{\text{un},P_{i,\text{nuc}}}$ rate constant of the assembly of new vRNP (h^{-1})

$k_{\text{un},S_{\text{un,nuc}}}$ rate constant of the assembly of new vRNP (h^{-1})

k_{va} specific virus attachment rate (mL h^{-1})

$k_{\text{v-c}}$ rate coefficient of cRNA synthesis (h^{-1})

k_{vd} specific virus degradation rate (h^{-1})

$k_{\text{v-degr}}$ rate constant of vRNA degradation (h^{-1})

k_{vi} specific virus infection rate (mL h^{-1})

$k_{\text{v-vm},i}$	rate coefficient of vmRNA synthesis (h^{-1})
$k_{\text{vm},i,\text{cyt-degr}}$	rate constant of cytoplasmic vmRNA degradation (h^{-1})
$k_{\text{vm},i,\text{nuc-cyt}}$	rate coefficient of vmRNA nuclear export (h^{-1})
$k_{\text{vm},i,\text{nuc-degr}}$	rate constant of nuclear vmRNA degradation (h^{-1})
N_{cells}	number of cells (cells)
P_{cell}	number cellular amino acids (amino acids cell^{-1})
$P_{i,\text{cyt}}$	number of the *i*-th protein in the cytoplasm (amino acids cell^{-1})
$P_{i,\text{nuc}}$	number of the *i*-th protein in the nucleus (amino acids cell^{-1})
$P_{j,\text{bud}}$	number of the *j*-th protein at the budding site (amino acids cell^{-1})
$P_{j,\text{ER}}$	number of the *j*-th protein in the ER (amino acids cell^{-1})
$P_{\text{NP,nuc}}$	number of NP proteins in the nucleus (amino acids cell^{-1})
$P_{\text{Pol,nuc}}$	number of polymerase complexes in the nucleus (amino acids cell^{-1})
R	number of cellular ribosomes (ribosomes cell^{-1})
R_{sf}	number of cellular receptors (receptors cell^{-1})
S_{cyt}	number of vRNPs in the cytoplasm (vRNPs cell^{-1})
S_{nuc}	number of vRNPs in the nucleus (vRNPs cell^{-1})
$S_{\text{un,bud}}$	newly synthesized vRNPs at the budding site (vRNPs cell^{-1})
$S_{\text{un,nuc}}$	newly synthesized vRNPs in the nucleus (vRNPs cell^{-1})
t	time (h)
U_C	concentration of uninfected cells (cells mL^{-1})
U_{r}	volume of medium containing N_{cells} cells *(nL)*
V	concentration of virus particles (virions mL^{-1})
V_{bud}	number of budding viruses (virions cell^{-1})
V_{end}	number of endosomal viruses (virions cell^{-1})
V_{ex}	number of extracellular viruses (virions nL^{-1})
V_{rel}	number of released viruses (virions nL^{-1})

V_s	number of surface viruses (virions cell^{-1})

Greek

α	rise rate (phages min^{-1})
λ	latent period (min)
μ	specific cell growth rate (h^{-1})
μ_max	maximum specific cell growth rate (h^{-1})
μ_vir	specific virus replication rate (h^{-1})
ν	eclipse period (min)
τ	time delay (h)
ω_i	fraction of *i*-th mRNA nucleotides (–)

References

1 Alberts, B., A. Johnson, J. Lewis, M. Raff, K. Roberts and P. Walter. 2002. *Molecular Biology of the Cell*, 4th edn. Garland, New York, NY.

2 Bailey, J. E. 1998. Mathematical modeling and analysis in biochemical engineering: past accomplishments and future opportunities. Biotech. Prog. **14**: 8–20.

3 Bremer, H. and P. P. Dennis. 1996. Modulation of chemical composition and other parameters of the cell by growth rate. In Neidhardt, F. C., R. Curtiss, III, C. Gross, et al. (eds.), *Escherichia coli and Salmonella: Cellular and Molecular Biology*. American Society for Microbiology, Washington, DC: 1553–69.

4 Büchen-Osmond, C. 2003. *Taxonomy and Classification of Viruses, Manual of Clinical Microbiology*, 8th edn, vol. 2. ASM Press, Washington, DC: 1217–26.

5 Buchholtz, F. and F. W. Schneider. 1987. Computer simulation of T3/T7 phage infection using lag times. Biophys. Chem. **26**: 171–9.

6 Chen, W., P. A. Calvo, D. Malide, et al. 2001. A novel influenza A virus mitochondrial protein that induces cell death. Nat. Med. **7**: 1306–12.

7 Dee, K. U., D. A. Hammer and M. L. Shuler. 1995. A model of the binding, entry, uncoating, and RNA synthesis of Semliki forest virus in baby hamster kidney (BHK-21) cells. Biotechnol. Bioeng **46**: 485–96.

8 Dee, K. U. and M. L. Shuler. 1997. A mathematical model of the trafficking of acid-dependent enveloped viruses: application to the binding, uptake, and nuclear accumulation of baculovirus. Biotechnol. Bioeng. **54**: 468–90.

9 Dulbecco, R. 1952. Production of plaques in monolayer tissue cultures by single particles of an animal virus. Proc. Natl Acad. Sci. USA **38**: 747–52.

10 Ellis, E. L. and M. Delbrück. 1939. The growth of bacteriophage. J. Gen. Physiol. **22**: 365–84.

11 Enami, M., G. Sharma, C. Benham and P. Palese. 1991. An influenza virus containing nine different RNA segments. Virology **185**: 291–8.

12 Enden, G., Y. H. Zhang and J. C. Merchuk. 2005. A model of the

dynamics of insect cell infection at low multiplicity of infection. J. Theor. Biol. **237**: 257–64.

13 ENDY, D., D. KONG AND J. YIN. 1997. Intracellular kinetics of a growing virus: a genetically structured simulation for bacteriophage T7. Biotechnol. Bioeng. **55**: 375–89.

14 FIELDS, S. AND M. JOHNSTON. 2005. Cell biology. Whither model organism research? Science **307**: 1885–6.

15 FLINT, J. S., L. W. ENQUIST, V. R. RACANIELLO, R. KRUG AND A. M. SALKA. 2000. *Principles of Virology: Molecular Biology, Pathogenesis, and Control*. ASM Press, Washington, DC.

16 FREDRICKSON, A. G., R. D. MEGEE, III AND H. M. TSUCHIYA. 1970. Mathematical models for fermentation processes. Adv. Appl. Microbiol. **13**: 419–65.

17 FUJII, Y., H. GOTO, T. WATANABE, T. YOSHIDA AND Y. KAWAOKA. 2003. Selective incorporation of influenza virus RNA segments into virions. Proc. Natl Acad. Sci. USA **100**: 2002–7.

18 GEISS, G. K., M. SALVATORE, T. M. TUMPEY, et al. 2002. Cellular transcriptional profiling in influenza A virus-infected lung epithelial cells: the role of the nonstructural NS1 protein in the evasion of the host innate defense and its potential contribution to pandemic influenza. Proc. Natl Acad. Sci. USA **99**: 10736–41.

19 GENBANK. http://www.ncbi.nih.gov/Genbank.

20 GENZEL, Y., I. BEHRENDT, S. KÖNIG, H. SANN AND U. REICHL. 2004. Metabolism of MDCK cells during cell growth and influenza virus production in large-scale microcarrier culture. Vaccine **22**: 2202–8.

21 GILCHRIST, M. A., D. COOMBS AND A. S. PERELSON. 2004. Optimizing within-host viral fitness: infected cell lifespan and virion production rate. J. Theor. Biol. **229**: 281–8.

22 GÜNTHER-AUSBORN, S., P. SCHOEN, I. BARTOLDUS, J. WILSCHUT AND T. STEGMANN. 2000. Role of hemagglutinin surface density in the initial stages of influenza virus fusion: lack of evidence for cooperativity. J. Virol. **74**: 2714–20.

23 HADAS, H., M. EINAV, I. FISHOV AND A. ZARITSKY. 1997. Bacteriophage T4 development depends on the physiology of its host *Escherichia coli*. Microbiology **134**: 179–85.

24 HASELTINE, E. L., J. B. RAWLINGS AND J. YIN. 2005. Dynamics of viral infections: incorporating both the intracellular and extracellular levels. Comput. Chem. Eng. **29**: 675–86.

25 JANG, J. D., C. S. SANDERSON, L. C. L. CHAN, J. P. BARFORD AND S. REID. 2000. Structured modeling of recombinant protein production in batch and fed-batch culture of baculovirus-infected insect cells. Cytotechnology **34**: 71–82.

26 KAUFMAN, R. J. 2000. Overview of vector design for mammalian gene expression. Mol. Biotechnol. **16**: 151–60.

27 KIM, H. AND J. YIN. 2004. Energy-efficient growth of phage Q Beta in *Escherichia coli*. Biotechnol. Bioeng. **88**: 148–56.

28 KRUEGER, A. P. AND J. H. NORTHROP. 1930. The kinetics of the bacterium–bacteriophage reaction. J. Gen. Physiol. **14**: 223–54.

29 KRUG, R. M., W. YUAN, D. L. NOAH AND A. G. LATHAM. 2003. Intracellular warfare between human influenza viruses and human cells: the roles of the viral NS1 protein. Virology **309**: 181–9.

30 LAMB, R. A. AND P. W. CHOPPIN. 1976. Synthesis of influenza virus proteins in infected cells: translation of viral polypeptides, including three P polypeptides, from RNA produced by primary transcription. Virology **74**: 504–19.

31 LAMB, R. A. AND R. M. KRUG. 2001. Orthomyxoviridae: the viruses and their replication. In KNIPE, D. M. AND P. M. HOWLEY (eds.), *Field's Virology*, 4th edn. Lippincott Williams & Wilkins, Philadelphia, PA: 1487–579.

32 LICARI, P. AND J. E. BAILEY. 1992. Modeling the population dynamics of baculovirus-infected insect cells: Optimizing infection strategies for enhanced recombinant protein yields. Biotechnol. Bioeng. **39**: 432–41.

33 LIN, C., R. E. HOLLAND, JR., J. C. DONOFRIO, M. H. MCCOY, L. R. TUDOR AND T. M. CHAMBERS. 2002. Caspase activation in equine influenza virus induced apoptotic cell death. Vet. Microbiol. **84**: 357–65.

34 LODISH, H., A. BERK, P. MATSUDAIRA, A. KAISER, M. KRIEGER, M. P. SCOTT, L. ZIPURSKY AND J. DARNELL. 2003. *Molecular Cell Biology*, 5th edn. Freeman, New York, NY.

35 LOWY, R. J. 2003. Influenza virus induction of apoptosis by intrinsic and extrinsic mechanisms. Int. Rev. Immunol. **22**: 425–49.

36 LUDWIG, S., S. PLESCHKA AND T. WOLFF. 1999. A fatal relationship – influenza virus interactions with the host cell. Viral Immunol. **12**: 175–96.

37 MARTIN, K. AND A. HELENIUS. 1991. Transport of incoming influenza virus nucleocapsids into the nucleus. J. Virol. **65**: 232–44.

38 MITTAL, A. AND J. BENTZ. 2001. Comprehensive kinetic analysis of influenza hemagglutinin-mediated membrane fusion: role of sialate binding. Biophys. J. **81**: 1521–35.

39 MÖHLER, L., D. FLOCKERZI, H. SANN AND U. REICHL. 2005. Mathematical model of influenza A virus production in large-scale microcarrier culture. Biotechnol. Bioeng. **90**: 46–58.

40 NELSON, D. L. AND M. M. COX. 2004. *Lehninger: Principles of Biochemistry*, 4th edn. Freeman, New York, NY.

41 NELSON, P. W. AND A. S. PERELSON. 2002. Mathematical analysis of delay differential equation models of HIV-1 infection. *Math Biosci.* **179**: 73–94.

42 NICHOLSON, K. G., R. G. WEBSTER AND A. J. HAY. 1998. *Textbook of Influenza*. Blackwell Science, Oxford.

43 NIELSEN, L. K. 2000. Virus production from cell culture kinetics. In SPIER, R. E. (ed.), *Encyclopedia of Cell Technology*, vol. 2. Wiley, New York, NY: 1217–30.

44 NOWAK, M. A. AND R. M. MAY. 2000. *Virus Dynamics: Mathematical Principles of Immunology and Virology*. Oxford University Press, Oxford.

45 NUNES-CORREIA, I., J. RAMALHO-SANTOS, S. NIR AND M. C. PEDROSO DE LIMA. 1999. Interactions of influenza virus with cultured cells: detailed kinetic modeling of binding and endocytosis. Biochemistry **38**: 1095–101.

46 PARK, Y. W. AND M. G. KATZE. 1995. Translational control by influenza virus. Identification of *cis*-acting sequences and *trans*-acting factors which may regulate selective viral mRNA translation. *J. Biol. Chem.* **270**: 28433–39.

47 RABINOVITCH, A., I. FISHOV, H. HADAS, M. EINAV AND A. ZARITSKY. 2002. Bacteriophage T4 development in *Escherichia coli* is growth rate dependent. J. Theor. Biol. **216**: 1–4.

48 RABINOVITCH, A., H. HADAS, M. EINAV, Z. MELAMED AND A. ZARITSKY. 1999. Model for bacteriophage T4 development in *Escherichia coli*. J. Bacteriol. **181**: 1677–83.

49 RABINOVITCH, A., A. ZARITSKY, I. FISHOV, M. EINAV AND H. HADAS. 1999. Bacterial lysis by phage – a theoretical model. J. Theor. Biol. **201**: 209–13.

50 REDDY, B. AND Y. YIN. 1999. Quantitative intracellular kinetics of HIV type 1. Aids Res. Human Retrovir. **15**: 273–83.

51 SCHULTZ-CHERRY, S., M. KOCI, E. THOMPSON AND T. M. TUMPEY. 2003. Examining the cellular pathways involved in influenza virus induced apoptosis. Avian Dis. **47**: 968–71.

52 SIDORENKO, Y. AND U. REICHL. 2004. Structured model of influenza virus replication in MDCK cells. Biotechnol. Bioeng. **88**: 1–14.

53 SIECZKARSKI, S. B. AND G. R. WHITTAKER. 2002. Dissecting virus entry via endocytosis. J. Gen. Virol. **83**: 1535–45.

54 SIECZKARSKI, S. B. AND G. R. WHITTAKER. 2002. Influenza virus can enter and infect cells in the absence of clathrin-mediated endocytosis. J. Virol. **76**: 10455–64.

55 SRIVASTAVA, R., L. YOU, J. SUMMERS AND J. YIN. 2002. Stochastic vs. deterministic modeling of intracellular viral kinetics. J. Theor. Biol. **218**: 309–21.

56 STOUTHAMER, A. H. 1979. The search or correlation between theoretical and experimental growth yields. In

QUAYLE, R. R. (ed.), *International Review Of Biochemistry: Microbial Biochemistry*. University Park Press, Baltimore MD: 1–47.

57 TAKIZAWA, T., C. TATEMATSU, K. OHASHI AND Y. NAKANISHI. 1999. Recruitment of apoptotic cysteine proteases (caspases) in influenza virus-induced cell death. Microbiol. Immunol. **43**: 245–52.

58 TSUCHIYA, H. M., A. G. FREDRICKSON AND R. ARIS. 1966. Dynamics of microbial cell populations. Adv. Chem. Eng. **6**: 125–206.

59 WHITTAKER, G. R., M. BUI AND A. HELENIUS. 1996. The role of nuclear import and export in influenza virus infection. Trends Cell. Biol. **6**: 67–71.

60 WHO. 2004. Communicable Disease Surveillance & Response (CSR), Estimating the impact of the next influenza pandemic: enhancing preparedness. http://www.who.int/csr/disease/influenza/en.

61 WHO. 2003. Influenza. http://www.who.int/mediacentre/-factsheets/fs211/en.

62 WICKHAM, T. J., R. R. GRANADOS, H. A. WOOD, D. A. HAMMER AND M. L. SHULER. 1990. General analysis of receptor-mediated viral attachment to cell surfaces. Biophys. J. **58**: 1501–16.

63 YAMANAKA, K., A. ISHIHAMA AND K. NAGATA. 1988. Translational regulation of influenza virus mRNAs. Virus Genes **2**: 19–30.

64 YOU, L., P. F. SUTHERS AND J. YIN. 2002. Effects of *Escherichia coli* physiology on growth of phage T7 *in vivo* and *in silico*. J. Bacteriol. **184**: 1888–94.

65 YOU, L. AND J. YIN. 2001. Simulating the growth of viruses. Pac. Symp. Biocomput.: 532–43.

66 ZHIRNOV, O. P., T. E. KONAKOVA, W. GARTEN AND H. KLENK. 1999. Caspase-dependent N-terminal cleavage of influenza virus nucleocapsid protein in infected cells. J. Virol. **73**: 10158–63.

Appendix

Model equations

The model is represented by a system of nonlinear ODEs, which represent the key steps of the influenza A virus infection cycle (Figure 4).

1 Virus Entry into the Host Cell

$$\frac{\mathrm{d}V_{\mathrm{ex}}}{\mathrm{d}t} = -k_{\text{ex-s},V_{\mathrm{ex}}} V_{\mathrm{ex}} + k_{\text{s-ex},V_{\mathrm{ex}}} V_{\mathrm{s}} \tag{1.1}$$

$$\frac{\mathrm{d}V_{\mathrm{s}}}{\mathrm{d}t} = k_{\text{ex-s},V_{\mathrm{s}}} V_{\mathrm{ex}} - k_{\text{s-ex},V_{\mathrm{s}}} V_{\mathrm{s}} - k_{\text{s-end}} V_{\mathrm{s}} \tag{1.2}$$

$$\frac{\mathrm{d}V_{\mathrm{end}}}{\mathrm{d}t} = k_{\text{s-end}} V_{\mathrm{s}} - k_{\text{end-cyt},V_{\mathrm{end}}} V_{\mathrm{end}} - k_{\text{end-degr}} V_{\mathrm{end}} \tag{1.3}$$

and

$$k_{\text{ex-s},V_{\mathrm{s}}} = \frac{U_{\mathrm{r}}}{N_{\mathrm{cells}}} k_{\text{ex-s},V_{\mathrm{ex}}}$$

$$k_{\text{s-ex},V_{\mathrm{s}}} = \frac{U_{\mathrm{r}}}{N_{\mathrm{cells}}} k_{\text{s-ex},V_{\mathrm{ex}}} \,.$$

2 vRNP Uncoating and Transport into the Nucleus

$$\frac{\mathrm{d}S_{\text{cyt}}}{\mathrm{d}t} = k_{\text{end-cyt},S_{\text{cyt}}} V_{\text{end}} - k_{\text{cyt-nuc}} S_{\text{cyt}} \tag{2.1}$$

$$\frac{\mathrm{d}S_{\text{nuc}}}{\mathrm{d}t} = k_{\text{cyt-nuc}} S_{\text{cyt}} - k_{\text{spl},S_{\text{nuc}}} S_{\text{nuc}} \tag{2.2}$$

and:

$$k_{\text{end-cyt},S_{\text{cyt}}} = N_{\text{seg}} k_{\text{end-cyt},V_{\text{end}}} \,. \tag{2.3}$$

3 Transcription [vmRNA(+) Production]

$$\frac{\mathrm{d}C_{i,\text{nuc}}}{\mathrm{d}t} = k_{\text{v-vm},i} P_{\text{Pol,nuc}} - k_{\text{vm},i,\text{nuc-cyt}} C_{i,\text{nuc}} - k_{\text{vm},i,\text{nuc-degr}} C_{i,\text{nuc}} \tag{3.1}$$

$$\frac{\mathrm{d}C_{i,\text{cyt}}}{\mathrm{d}t} = k_{\text{vm},i,\text{nuc-cyt}} C_{i,\text{nuc}} - k_{\text{vm},i,\text{cyt-degr}} C_{i,\text{cyt}} \tag{3.2}$$

$i \in [\text{Pol, NP, M1, NS1, NS2, M2, HA, NA}]$,

with the following assumption concerning the kinetic coefficients:

$$k_{\text{v-vm},i} = k_{\text{v-vm},i,\text{max}} \frac{1}{1 + a_{\text{NP}} P_{\text{NP,nuc}}} \,.$$

4 Viral Genome Replication

$$\frac{\mathrm{d}C_{\text{c}}}{\mathrm{d}t} = k_{\text{v-c}} P_{\text{Pol,nuc}} - k_{\text{c-degr}} C_{\text{c}} \tag{4.1}$$

$$\frac{\mathrm{d}C_{\text{v}}}{\mathrm{d}t} = k_{\text{c-v}} P_{\text{Pol,nuc}} + k_{\text{spl},C_{\text{v}}} S_{\text{nuc}} - k_{\text{un},C_{\text{v}}} C_{\text{v}} \prod_{l} P_{l,\text{nuc}} - k_{\text{v-degr}} C_{\text{v}} \tag{4.2}$$

$i \in [\text{Pol, NP, M1, NS2}]$,

with the following assumptions concerning the kinetic coefficients:

$$k_{\text{v-c}} = k_{\text{v-c,max}} \frac{P_{\text{NP,nuc}}}{b_{\text{NP}} + P_{\text{NP,nuc}}}$$

$$k_{\text{c-v}} = k_{\text{c-v,max}} \frac{P_{\text{NP,nuc}}}{b_{\text{NP}} + P_{\text{NP,nuc}}}$$

$$k_{\text{spl},C_{\text{v}}} = C_{\text{seg}} k_{\text{spl},S_{\text{nuc}}} \,.$$

5 Capsid, Nonstructural and Matrix Protein Production

$$\frac{\mathrm{d}P_{i,\text{cyt}}}{\mathrm{d}t} = k_{i,\text{synt}} \frac{C_{i\text{cyt}}}{\mathrm{d}_{\text{rib}}} - k_{i,\text{cyt-nuc}} P_{i,\text{cyt}} - k_{i,\text{cyt-degr}} P_{i,\text{cyt}} \tag{5.1}$$

$$\frac{\mathrm{d}P_{i,\mathrm{nuc}}}{\mathrm{d}t} = k_{i,\mathrm{cyt\text{-}nuc}}P_{i,\mathrm{cyt}} + k_{\mathrm{spl},P_{i,\mathrm{cyt}}}S_{\mathrm{nuc}} - k_{\mathrm{un},P_{i,\mathrm{nuc}}}C_{\mathrm{v}}\prod_{l}P_{l,\mathrm{nuc}} - k_{i,\mathrm{nuc\text{-}degr}}P_{i,\mathrm{nuc}} \tag{5.2}$$

$i \in [\mathrm{Pol, NP, M1, NS1, NS2}]$ $l \in[\mathrm{Pol, NP, M1, NS2}]$,

and:

$k_{\mathrm{spl},P_{i,\mathrm{nuc}}} = P_{i,\mathrm{seg}}k_{\mathrm{spl}S_{\mathrm{nuc}}}$

$k_{i,\mathrm{synt}} = k_{\mathrm{Rib}}$ (k_{Rib} rate of peptide chain elongation, Table A2).

6 Envelope Protein Production

$$\frac{\mathrm{d}P_{j,\mathrm{ER}}}{\mathrm{d}t} = k_{j,\mathrm{synt}}\frac{C_{j,\mathrm{cyt}}}{\mathrm{d}_{\mathrm{Rib}}} - k_{j,\mathrm{Er\text{-}bud}}P_{j,\mathrm{ER}} - k_{j,\mathrm{ER\text{-}degr}}P_{j,\mathrm{ER}} \tag{6.1}$$

$$\frac{\mathrm{d}P_{j,\mathrm{bud}}}{\mathrm{d}t} = k_{j,\mathrm{ER\text{-}}bud}P_{j,\mathrm{ER}} - k_{\mathrm{bud},P_{j,\mathrm{bud}}}S_{\mathrm{un,bud}}\prod_{l}P_{l,\mathrm{bud}} - k_{j,\mathrm{bud\text{-}degr}}P_{j,\mathrm{bud}} \tag{6.2}$$

$j, l \in [\mathrm{M2, HA, NA}]$

7 Packaging

$$\frac{\mathrm{d}S_{\mathrm{un,nuc}}}{\mathrm{d}t} = k_{\mathrm{un},S_{\mathrm{un,nuc}}}C_{\mathrm{v}}\prod_{l}P_{l,\mathrm{nuc}} - k_{\mathrm{un,nuc\text{-}bud}}S_{\mathrm{un,nuc}} \tag{7.1}$$

$$\frac{\mathrm{d}S_{\mathrm{un,bud}}}{\mathrm{d}t} = k_{\mathrm{un,nuc\text{-}bud}}S_{\mathrm{un,nuc}} - k_{\mathrm{bud},S_{\mathrm{un,bud}}}S_{\mathrm{un,bud}}\prod_{s}P_{s,\mathrm{bud}} \tag{7.2}$$

$l \in [\mathrm{Pol, NP, M1, NS2}]$ $s \in [\mathrm{M2, HA, NA}]$,

and:

$k_{\mathrm{un},C_{\mathrm{v}}} = C_{\mathrm{seg}}k_{\mathrm{un},S_{\mathrm{un,nuc}}}$

$k_{\mathrm{un}P_{i,\mathrm{nuc}}} = P_{i,\mathrm{seg}}k_{\mathrm{un},S_{\mathrm{un,nuc}}}$.

8 Virus Budding and Release

$$\frac{\mathrm{d}V_{\mathrm{bud}}}{\mathrm{d}t} = k_{\mathrm{bud},V_{\mathrm{bud}}}S_{\mathrm{un,bud}}\prod_{l}P_{l,\mathrm{bud}} - k_{\mathrm{bud\text{-}rel},V_{\mathrm{bud}}}V_{\mathrm{bud}} \tag{8.1}$$

$l \in [\mathrm{M2, HA, NA}]$

$$\frac{\mathrm{d}V_{\mathrm{rel}}}{\mathrm{d}t} = k_{\mathrm{bud\text{-}rel},V_{\mathrm{rel}}}V_{\mathrm{bud}} , \tag{8.2}$$

and:

$k_{\mathrm{bud},P_{j,\mathrm{bud}}} = P_{j,\mathrm{vir}}k_{\mathrm{bud},V_{\mathrm{bud}}}$

$k_{\mathrm{bud},S_{\mathrm{un,bud}}} = N_{\mathrm{seg}}k_{\mathrm{bud},V_{\mathrm{bud}}}$

$k_{\mathrm{bud\text{-}rel},V_{\mathrm{bud}}} = \frac{U_{\mathrm{r}}}{N_{\mathrm{cells}}}k_{\mathrm{bud\text{-}rel},V_{\mathrm{rel}}}$

Initial Conditions and Kinetic Parameters

Initial conditions, kinetic parameters and assumptions are summarized in the following two tables. ODEs were solved using MATLAB version 6.5 release 13 (Mathworks, Natick, MA, USA).

Table A1 Basic assumptions on host cell and virus composition used for simulation studies of the structured mathematical model of the influenza A virus life cycle (Section 4.3)

Parameter	Value	Source
Host cell		
number of receptors (receptors cell^{-1})	10^4–10^5	[62]
number of endosomes (endosomes cell^{-1})	200	[1]
number of ribosomes (ribosomes cell^{-1})	5×10^6	[1]
distance between ribosomes on mRNA (nucleotides)	80	[1]
number of nuclear pores (pores cell^{-1})	3000–4000	[15]
dry weight of a host cell (ng cell^{-1})	0.54	–[a]
number of free nucleotides (nucleotides cell^{-1})	1.3×10^{10}	[1, 26]
number of nuclear precursor mRNAs (molecules cell^{-1})	2.2×10^5	[1, 26]
average number of nucleotides per mRNA (nucleotides)	6000	[1, 26]
number of free amino acids (amino acids cell^{-1})	3.1×10^{10}	[1, 40]
Virus (influenza A, A/PR/8/34)		
number of genome segments (segments virion^{-1})	8	[31]
full length of the genome (nucleotides virion^{-1})	13588	[31]
average length of one genome segment (nucleotides)	1699	[31]
total number of amino acids (amino acids virion^{-1})	2.4×10^6	[31]

[a] For MDCK cells (own data).

Table A2 Kinetic parameters used for simulation studies of the structured mathematical model of the influenza A virus life cycle (Section 4.3)

<table>
<tr><th colspan="2">Parameter</th><th>Value</th><th>Source</th></tr>
<tr><td colspan="4">Rate coefficients from literature</td></tr>
<tr><td colspan="2">Rate constant for endocytosis $k_{\text{s-end}}$ (s^{-1})</td><td>2.6×10^{-4}</td><td>[45]</td></tr>
<tr><td colspan="2">Rate of peptide chain elongation k_{Rib} (s^{-1})</td><td>5.0</td><td>[1]</td></tr>
<tr><td colspan="2">Rate of RNA synthesis k_{Pl} (s^{-1})</td><td>30.0</td><td>[1]</td></tr>
<tr><td colspan="4">Estimated rate coefficients</td></tr>
<tr><th>Parameter</th><th>Value</th><th>Parameter</th><th>Value</th></tr>
<tr><td>$k_{\text{ex-s},V_\text{s}}$ (h^{-1})</td><td>8.4[a]</td><td>$k_{\text{c-degr}}$ (h^{-1})</td><td>100.0</td></tr>
<tr><td>$k_{\text{s-ex},V_\text{s}}$ (h^{-1})</td><td>4.444[a]</td><td>$k_{\text{v-degr}}$ (h^{-1})</td><td>100.0</td></tr>
<tr><td>$k_{\text{s-end}}$ (h^{-1})</td><td>0.936</td><td>$k_{i,\text{cyt-nuc}}$ (h^{-1})</td><td>1.0</td></tr>
<tr><td>$k_{\text{end}-\text{cyt},V_\text{end}}$ (h^{-1})</td><td>14.0</td><td>$k_{i,\text{cyt-degr}}$ (h^{-1})</td><td>0.01</td></tr>
<tr><td>$k_{\text{end-degr}}$ (h^{-1})</td><td>6.0</td><td>$k_{i,\text{nuc-degr}}$ (h^{-1})</td><td>5.0</td></tr>
<tr><td>$k_{\text{cyt-nuc}}$ (h^{-1})</td><td>5.0</td><td>$k_{j,\text{ER-}bud}$ (h^{-1})</td><td>1.0</td></tr>
<tr><td>$k_{\text{spl},S_\text{nuc}}$ (h^{-1})</td><td>1.0</td><td>$k_{j,\text{Er-degr}}$ (h^{-1})</td><td>0.01</td></tr>
<tr><td>$k_{\text{vm},i,\text{nuc-cyt}}$ (h^{-1})</td><td>1.0</td><td>$k_{j,\text{bud-degr}}$ (h^{-1})</td><td>5.0</td></tr>
<tr><td>$k_{\text{vm},i,\text{nuc-degr}}$ (h^{-1})</td><td>0.1</td><td>$k_{\text{un,nuc-bud}}$ (h^{-1})</td><td>1.0</td></tr>
<tr><td>$k_{\text{vm},i,\text{cyt-degr}}$ (h^{-1})</td><td>0.01</td><td>$k_{\text{bud-rel},V_\text{bud}}$ (h^{-1})</td><td>1.0</td></tr>
</table>

[45] Nunes-Correia et al., 1999, [1] Alberts et al., 2002

[a] For MDCK cells (own data).

Part 7 Analysis of Expression Data

24
DNA Microarray Technology and Applications – An Overview

John Quackenbush

1 Introduction to DNA Microarrays

The Human Genome Project promised to transform biology and medicine by providing us with a complete human genome sequence and a catalogue of all human genes. However, neither the 2001 announcement of completion of a draft human genome sequence by the competing public and private efforts to sequence the genome [50, 93] nor the 2004 publication of a "finished" human genome sequence [1] have delivered on that promise. Rather, genomics has transformed biology by providing technologies such as DNA microarrays, proteomics and metabolomics that are allowing us to generate holistic data on patters of gene, protein or metabolite expression. These approaches are seeing increasing applications in the study of human disease. There have been a number of promising studies in which microarray data, in particular, have been useful for discovering new molecular classes of previously well-studied diseases. The molecular fingerprints that emerge from such analyses can also be used for classification of disease, with the expression profiles, rather than the presence or levels of specific proteins, serving as biomarkers for diagnostic and prognostic classification.

2 Microarrays and Clinical Applications

In 1995, DNA microarrays were first reported as a tool for probing transcriptional levels on a genomic scale [54, 78] and the research community quickly seized upon this approach as a means of identifying genes that might provide

Bioinformatics – From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

insight into a wide range of biological processes. Most of the early experiments adopted as simple, yet powerful design – comparing two biological classes in order to identify genes that were differentially expressed between them. These experiments generally sought to gain insight into the underlying biology and microarrays were used as a tool for gene discovery; many of the early applications were to the analysis of gene expression in human cancers [25,46,97].

It did not take long, however, for many to realize that the utility of microarray-based approaches extended beyond mechanistic studies; arrays could be used to find new subclasses in disease states [3, 66], to identify new biomarkers that could be associated with disease [60] and even that the expression patterns themselves could be used as biomarkers to distinguish subclasses of disease [34]. This realization resulted in a proliferation of studies that searched for patterns of expression that could be used to classify tumor types [82], and to predict outcome [12,92] and response to chemotherapy [91].

Many of the earliest microarray publications on classification of cancer focused on cluster analysis of tumor samples and genes, including applications of hierarchical clustering [2,3,13,66,67,73,82] and partitioning methods such as self-organizing maps (SOMs) [34, 67]. These unsupervised data mining approaches have proven useful for class discovery as they take an unbiased approach to looking for patterns in the data to look for consistent subgroups in the data. Alizadeh and coworkers [2] used hierarchical clustering of cDNA microarray expression data from lymphoma samples to identify two previously unrecognized and transcriptionally distinct subclasses of diffuse large B cell lymphomas (DLBCLs) that were eventually related to different stages of B cell differentiation. One class expressed genes characteristic of germinal B cells (germinal center B-like DLBCL class), while the other expressed genes normally induced during *in vitro* activation of peripheral blood B cells (activated B-like DLBCL class). Their analysis also showed that patients within these subclasses had distinct clinical prognoses. Bhattacharjee and coworkers [13] used a similar approach to identify lung adenocarcinoma subclasses with different patient outcome. Hierarchical clustering of cDNA microarray expression data was also used by Ross and coworkers [73] to analyze the 60 cell lines form the National Cancer Institute's anticancer drug screening panel (the NCI 60 cell lines). Their analysis showed that cell line expression could be used to group samples based on their tissue of origin as well as to find genes that had similar patterns of expression across the samples.

However, in a clinical setting, the goal is generally not to look for new disease subtypes, but rather to use new techniques to provide better diagnostic and prognostic evidence to the clinician. In that light, the ability of gene expression profiles to distinguish different disease subtypes suggests that these same profiles can be used to classify samples based on the patterns of expression that are observed. Golub and coworkers [34] were the first to extend the class discovery approach to classification. Starting with expression profiles on acute lymphoblastic leukemia (ALL) and acute myeloblastic leukemia (AML) samples collected using Affymetrix GeneChips^{TM}. These samples were analyzed using SOMs and the tumor groups discovered based on expression patterns were compared to known classes. Their ability to partition the data into distinct ALL/AML groups prompted them to use the expression profiles as a means of classifying the samples. They proposed a weighted gene voting scheme for classification that is a variant on linear discriminant analysis methods. Since that time, there have been a large number of other studies describing classification approaches. Ramaswamy and colleagues [70] used support vector machines (SVMs) to demonstrate that expression profiles from microarrays could be used to separate tumors from 14 different organ sites, and Bloom and coworkers [14] used artificial neural networks (ANNs) to extend the approach to include 21 different tumor types profiled on both cDNA and Affymetrix GeneChips^{TM}, and further demonstrated that the same approach could be used to predict the primary source for metastatic lesions. Pomeroy and coworkers [67] profiled embryonic tumors of the central nervous system (CNS) on Affymetrix GeneChips^{TM}, and applied a range of unsupervised and supervised learning methods to investigate whether gene expression data could be used to distinguish between new and existing CNS tumor classes and for patient prognosis. Beer and coworkers [12] used an approach based on the Cox regression model to identify marker genes for predicting the survival of patients with lung adenocarcinoma.

Although mechanistic and discovery-based analyses of gene expression using microarrays continue to have widespread use, it is becoming increasingly clear that the use of array-based expression profiles for classification and class discovery will continue to be major applications for microarrays, as well as a broad range of other "omic" technologies, including proteomics, metabolomics and others. Fundamentally, all of these technologies can be used to ask a simple question: "Can we find a pattern that we can use to distinguish biological samples based on some inherent property?". To better understand how and when this question can be answered, it is useful to review some of the basic issues related to use of microarray data for the classification of human cancer and to point out some of the potential limitations of this approach.

3 Microarray Data Collection, Transformation and Representation

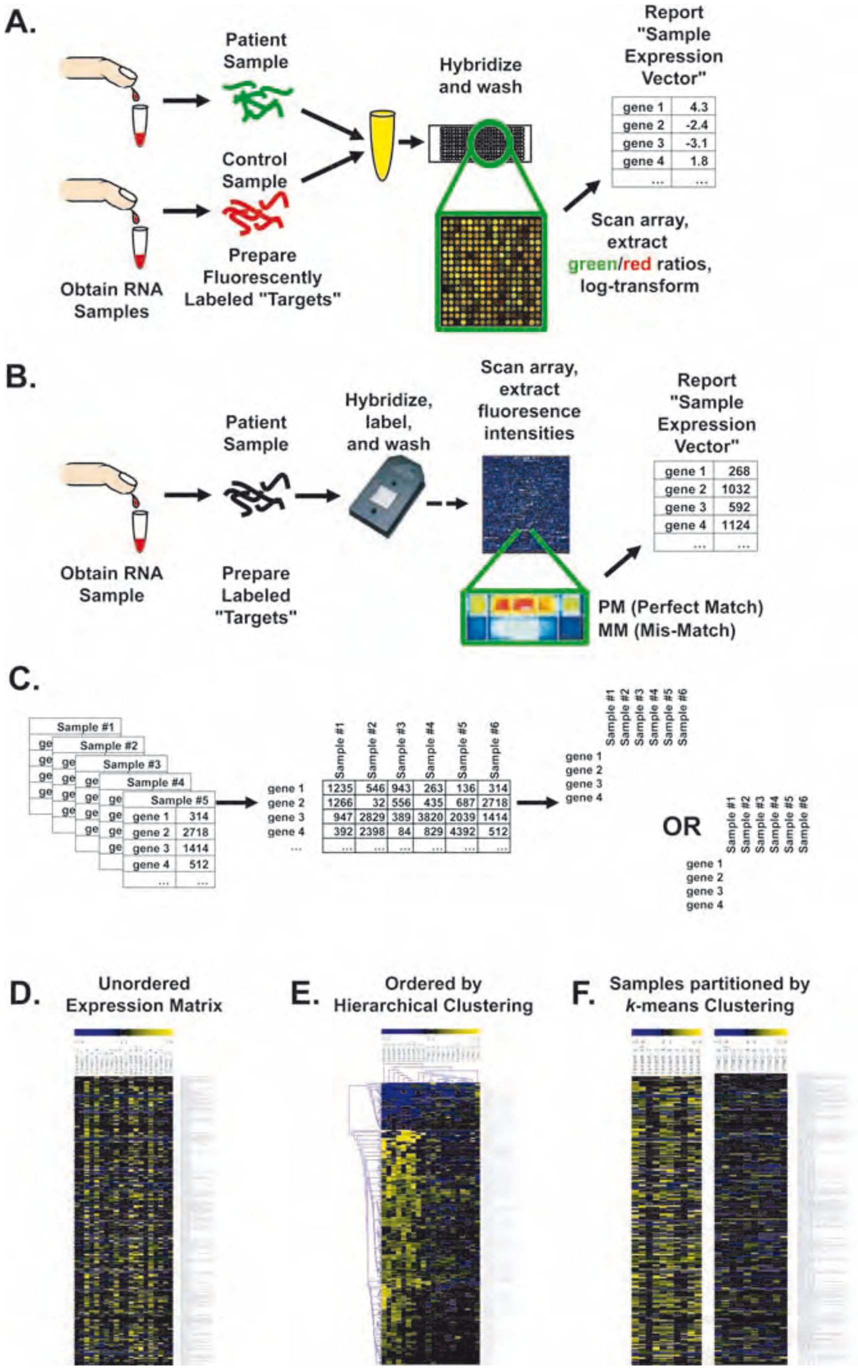
DNA microarray technology is relatively simple. Gene-specific probes, representing thousands of individual genes, are arrayed on an inert substrate (such as glass) and used to assay gene expression levels in selected biological samples. RNA is extracted from tissues of interest, converted into an appropriate target nucleic acid mixture labeled with a detectable marker (typically a fluorescent dye) and allowed to hybridize to the arrays. Individual RNA species hybridize to complementary gene-specific probes and hybridization is assayed by measuring fluorescence for each probe using a confocal laser scanner. These hybridization intensities, in turn, are used as estimates of gene-expression levels.

Microarray technology can be divided into two broad classes based on the way in which gene expression data is generated and how samples are compared. In a "two-color" microarray assay, two RNA samples, each labeled with a different dye, are simultaneously hybridized to a single array (Figure 1A). The sample of interest (or "query" sample), e.g. tissue from a cancerous colon tumor, is labeled with one dye and a control sample, e.g. normal colon mucosa, is labeled with another dye, and the two samples mixed in an approximate 1:1 ratio based on the quantity of labeled sample. Such as assay compares the expression levels between paired samples and reports expression levels as the logarithm of the ratio [the "$\log_2$(ratio)"] of the query to the control sample for each gene represented on the array. For single-color arrays, such as the widely-used Affymetrix GeneChip^{TM}, each sample is individually labeled and hybridized to a single array (Figure 1B). The expression for each gene is then reported as a single fluorescence intensity that represents an estimated expression level, less nonspecific background, although these measures can be used to derive others such as the $\log_2$(ratio) of expression levels in samples that one wishes to compare.

There are, of course, many variants of the various basic technologies and many commercial vendors. Two-color assays generally use either cDNA or long (50- to 70-mer) oligonucleotide probes while single-color assays typically are based on short (25-mer) or long (50- to 70-mer) oligonucleotide probes. Manufacturers of gene expression arrays include Affymetrix, Agilent, Illumina, Combimatrix, Nimblegen and a host of others, as well as "homebrew" arrays printed in individual laboratories. Labeling approaches use a variety of approaches including reverse transcription of RNA to cDNA (often used in two-color approaches), creation and fragmentation of antisense RNA (a protocol used on Affymetrix GeneChips), and various amplification approaches, most notably variations on Eberwine's T7 amplification. However, regardless of the approach or technology, the fundamental data used in all subsequent analyses are expression measures for each gene in each sample.

Following collection, the data are usually normalized to facilitate comparison between individual hybridization assays, and to compensate for differences in labeling, hybridization and detection efficiencies. There are several approaches to data normalization; the most appropriate approach depends on the type of array and assumptions regarding about biases in the data [40, 68, 76, 99, 100]. The data are then generally filtered using some set of objective criteria (e.g. eliminating those genes with minimal variance across samples or those with signals approaching background levels) or using some statistical analyses to select genes whose expression levels are correlated with particular groups of samples. These normalization and filtering transformations must be carefully applied as they can have a profound effect on the results obtained. Different statistical analysis methods applied to the very same data set can often produce different (but usually overlapping) sets of "significant" genes. Not surprisingly, the best way to deal with these "high-dimensional" data sets, in which there are often more measurements (genes) than samples, is an area of active research and debate. Chapter 25 gives a detailed account of current methods for preprocessing expression data. It should be noted that, to date, there is no single "right" way to analyze microarray data as even a single data set may reveal different features that are biologically relevant if a new analysis method is applied to the data. Extracting meaning from expression data really requires active participation of an expert in the biological system who is willing to explore the features contained within the data.

Once the normalized and filtered expression data are assembled, they are typically represented in an "expression matrix" in which each row represents a particular gene and each column represents a specific biological sample (Figure 1C). Each row is a "gene expression vector" where the individual entries are the expression levels of a specific gene across all of the samples and each column is a "sample expression vector" that records the expression of all genes in that sample. To ease interpretation of the results of multiple hybridizations, these data are often represented as a matrix in which individual elements are colored to indicate the expression level of each gene in each sample (Figure 1C), creating a visual representation of gene expression patterns across the collection of samples. The most common approach colors genes based on the $\log_2$(ratio) for each sample measured relative to some control, with $\log_2$(ratio) values close to zero colored black, those with $\log_2$(ratio) values greater than zero colored red (indicating "upregulated" genes) and those with negative values colored green (for "downregulated" genes), although other color schemes are kinder to those to those who are "red–green" colorblind (Figure 1C). The relative intensity of each element indicates the relative expression of the gene that it represents, with brighter elements indicating a higher level of expression. For any particular group of samples, the expression matrix generally appears without any apparent


A.
Obtain RNA Samples
Patient Sample
Control Sample
Prepare Fluorescently Labeled "Targets"
Hybridize and wash
Scan array, extract green/red ratios, log-transform
Report "Sample Expression Vector"
B.
Obtain RNA Sample
Patient Sample
Prepare Labeled "Targets"
Hybridize, label, and wash
Scan array, extract fluoresence intensities
PM (Perfect Match)
MM (Mis-Match)
Report "Sample Expression Vector"
C.
Sample #1
Sample #2
Sample #3
Sample #4
Sample #5
Sample #6
gene 1
gene 2
gene 3
gene 4
OR
D. Unordered Expression Matrix
E. Ordered by Hierarchical Clustering
F. Samples partitioned by k-means Clustering

gene 1	4.3
gene 2	-2.4
gene 3	-3.1
gene 4	1.8
...	...

gene 1	268
gene 2	1032
gene 3	592
gene 4	1124
...	...

Sample #5	
gene 1	314
gene 2	2718
gene 3	1414
gene 4	512
...	...

	Sample #1	Sample #2	Sample #3	Sample #4	Sample #5	Sample #6
gene 1	1235	546	943	263	136	314
gene 2	1266	32	556	435	687	2718
gene 3	947	2829	389	3820	2039	1414
gene 4	392	2398	84	829	4392	512
...	...	...	...	...	...	...

pattern or order (Figure 1D). Clustering programs generally reorder the rows, or columns, or both, such that patterns of expression become visually apparent (Figure 1E and F). While we will focus on using DNA microarray data, it should be noted that any data which can be placed into a "genes by samples" expression matrix format (e.g. "proteins by samples") can be analyzed using exactly the same techniques. This applies also to proteomics data discussed in Chapter 28, for instance.

4 Identifying Patterns of Expression

In doing microarray analysis, we are generally looking for genes that exhibit patterns of expression correlating with the biological states of the system being analyzed or that have "similar" patterns of expression across multiple samples; alternatively, we may look for samples that exhibit "similar" profiles. To facilitate this process, we need to define a measure of similarity between samples by defining distance measures. Although this might seem esoteric, there are a variety of distance measures that can be used and each can reveal different features in the data. Two of the most commonly used are Euclidean distance and Pearson correlation coefficient distances. Euclidean distance is important if the "magnitude" of expression levels is important as genes are scored as "close" if they are, for example, upregulated by same amount. On

←

Figure 1 An overview of DNA microarray analysis. (A) Single-color analysis, such as that using the Affymetrix GeneChip^{TM}, hybridizes labeled RNA from each biological sample to a single array in which a series of "perfect match" (PM) gene-specific probes are arrayed. Gene expression levels are estimated by measuring hybridization intensities are for each probe and background is measured using a corresponding set of "mismatch" (MM) probes. Gene expression levels are reported for each patient sample as a "sample expression vector" summarizing the difference between signal and background for each gene. (B) In two-color analysis approaches, RNA samples from patient and control samples are individually labeled with distinguishable fluorescent dyes and cohybridized to single DNA microarray consisting of individual gene-specific probes. Relative gene expression levels in the two samples are estimated by measuring the fluorescence intensities for each arrayed probe; a sample expression vector summarizing the expression level of each gene in the patient sample (relative to the control) is reported. (C) The collection of sample expression vectors is typically compiled into a single "expression matrix." Each column in the expression matrix represents an individual sample and its measured expression levels for each gene (the sample expression vector); each row represents a gene and its expression levels across all samples (a "gene expression vector"). The expression matrix is often visualized by presenting a colored matrix (typically red/green although other combinations such as blue/yellow are now common). Here, the color and its intensity represent the relative direction and magnitude of a gene expression difference. (D) An unordered data set, subjected to (E) average linkage hierarchical clustering or (F) k-means clustering reveals underlying patterns that can help identify classes in the data set; here the resulting two clusters are shown.

the other hand, Pearson correlation coefficients are useful if the "shape" of the expression profile is important so genes are close if their expression levels are increasing in some samples and decreasing in others, regardless of their absolute levels. Generally, in applications involving classification, we are interested in the pattern of expression across samples, so Pearson correlation coefficient distances are most useful, but not always.

Having recorded the appropriate data, normalized and filtered it, and chosen a means of measuring similarity, there are a variety of approaches we can take to looking for features and the methods we use are generally grouped into two broad classes: supervised and unsupervised methods. Supervised methods use information about the underlying structure of the data to search for patterns, e.g. using clinical classification to search for genes that correlate with class. These approaches are very useful when we know a great deal about the samples, but are not useful if we want to find new disease subclasses. Unsupervised methods are very useful for data exploration and can reveal unexpected patterns in the data, although the danger is that these patterns may be an artifact of the data or the algorithm itself.

5 Class Discovery

A useful first approach in the analysis of microarray data is to use an unsupervised method to explore expression patterns of that exist in the data. The question we are asking is: "Are there unexpected but biologically interesting patterns that exist in the data?". Unsupervised methods do not use the sample classification as input – they do not take into account, for example, whether the samples come from ALL or AML patients. They simply group samples together based on some measure of similarity between then. Two of the most widely used unsupervised approaches are hierarchical clustering and *k*-means clustering.

5.1 Hierarchical Clustering

Hierarchical clustering has become one of the most widely used techniques for the analysis of gene expression data; it has the advantage that it is simple and the result can be easily visualized [30, 59, 98]. Initially, we start with N clusters, where N is the number of genes (or samples) in the target data set. Hierarchical clustering (Figure 2) is an agglomerative approach in which single expression profiles are joined to form nodes, which are further joined until the process has been carried to completion, forming a single hierarchical tree. The algorithm proceeds in a straightforward manner for the clustering of genes (or similarly for samples):

(i) Calculate the pairwise distances for all of the genes to be clustered (using an appropriately selected distance measure).

(ii) Search all of the distances for the two most similar genes or clusters; initially each cluster consists of a single gene. This is the true first stage in the "clustering" process. If several pairs share the same similarity, a predetermined rule is used to decide between alternatives.

(iii) The two selected clusters are merged to produce a new cluster that now contains two or more objects.

(iv) The distances are calculated between this new cluster and all other clusters. There is no need to calculate *all* distances since only those involving the new cluster have changed.

(v) Steps (ii)–(iv) are repeated $N - 1$ times until all objects are in one cluster.

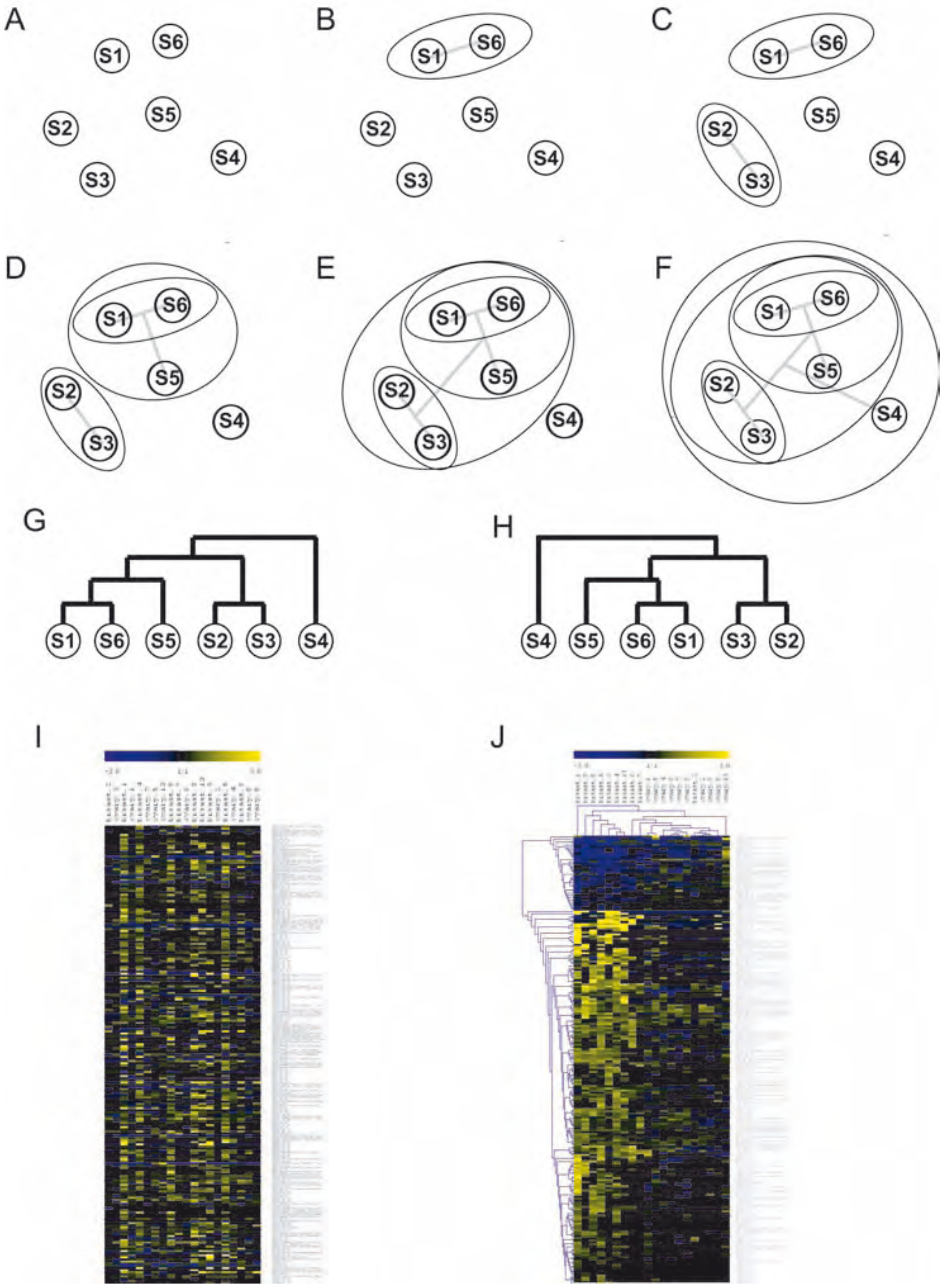
There are a number of variants of hierarchical clustering that reflect different approaches to calculating distances between the newly defined clusters and the other genes or clusters. *Single linkage* clustering determines the distance of two clusters to be the shortest distance between any pair of elements, one in each cluster, *complete linkage* clustering takes the largest distance between any such pair and *average linkage* clustering uses the average distance between all such pairs.

Typically, we represent the relationship between samples using a dendrogram where branches in the tree are built based on the connections determined between clusters as the algorithm progresses. In order to visualize the relationships between samples, we generally use the dendrogram to rearrange the columns (or rows as appropriate) in the expression matrix to better visualize patterns in the data set.

5.2 *k*-means Clustering

If there is prior knowledge regarding the number of clusters that should be represented in the data, *k*-means clustering (Figure 3) is a good alternative to hierarchical methods [5, 83]. In *k*-means, objects are partitioned into a fixed number (k) of clusters such that the clusters are internally similar but externally dissimilar; no dendrograms are produced (but one could use hierarchical techniques on each of the data partitions after they are constructed). The process involved in *k*-means clustering is conceptually simple, but can be computationally intensive:

(i) All initial objects (genes or samples) are randomly assigned to one of k clusters (where k is specified by the user).


A
S1
S6
S2
S5
S3
S4
B
C
D
E
F
G
H
I
J

(ii) An average expression vector is then calculated for each cluster and this is used to compute the distances between clusters.

(iii) Using an iterative method, objects are moved between clusters and intra- and inter-cluster distances are measured with each move. Objects are allowed to remain in the new cluster only if they are closer to it than to their previous cluster.

(iv) Following each move, the expression vectors for each cluster are recalculated.

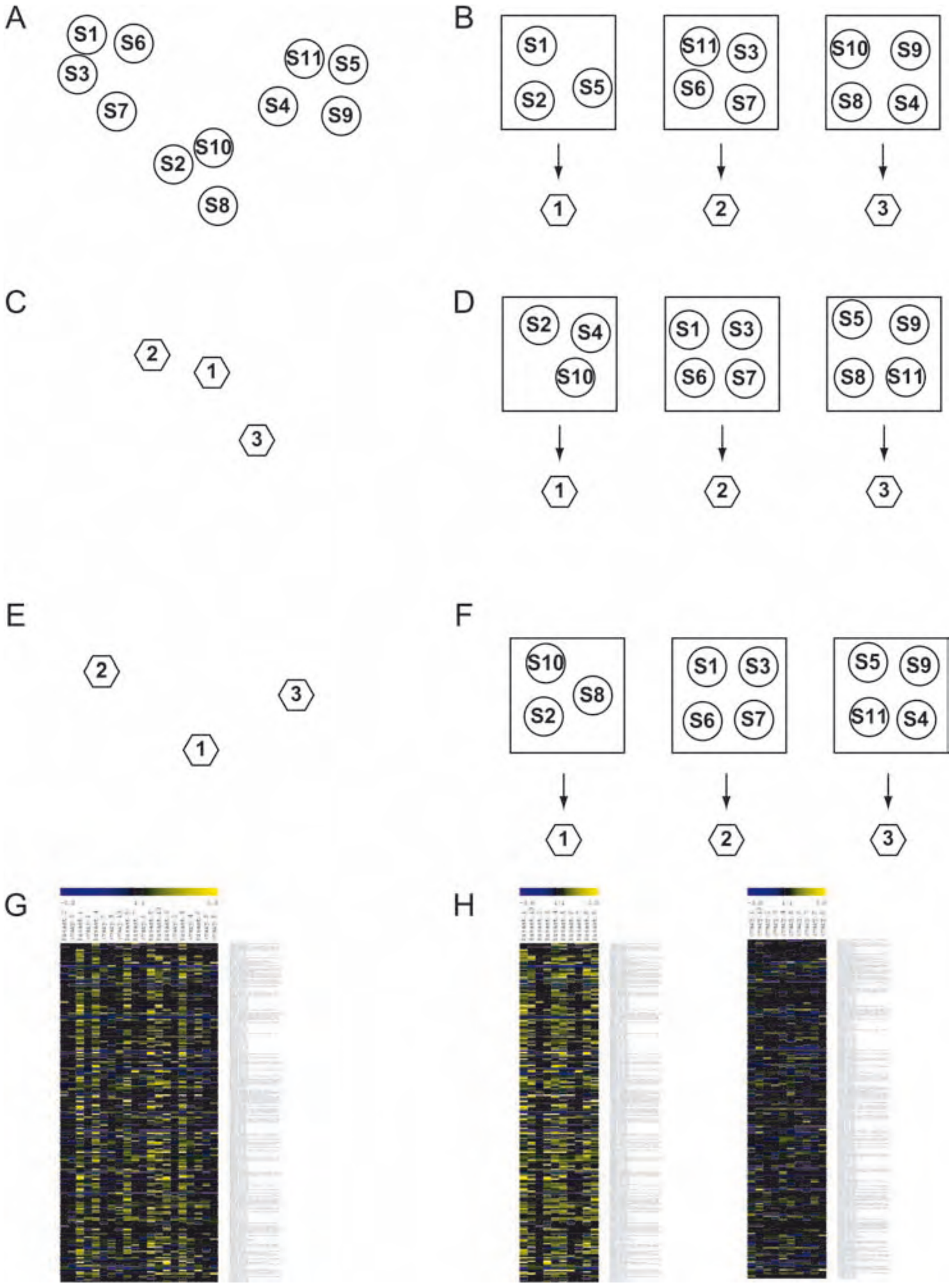
(v) The shuffling proceeds until moving any more objects would make the clusters more variable, increasing intra-cluster distances and decreasing inter-cluster dissimilarity.

Some implementations of *k*-means clustering allow not only the number of clusters to be specified, but also seed cases for each cluster. This has the potential to allow one to use prior knowledge of the system to help define the cluster output, such as a typical profile for a few key genes known to distinguish classes of patients. Of course, the "means" in *k*-means refers to the use of a mean expression vector for each emerging cluster. As one might imagine, there are variations also use other measures or each cluster, such as *k*-medians clustering.

Although *k*-means uses prior knowledge about the number of clusters, it is unsupervised in the sense that no prior knowledge about cluster membership is used in making the assignments. One can also use measures of cluster compactness to estimate the number of clusters in the data before beginning the process [102]. One important thing to remember about *k*-means clustering is that it is not deterministic – running the algorithm twice is likely to produce different clusters or associations. There are approaches to dealing with this ambiguity, e.g. running *k*-means multiple times at fixed *k* and using the consensus clusters.

←

Figure 2 Hierarchical clustering proceeds by (A) first collecting all of the objects (genes or samples) to be clustered and calculating the pairwise distance between them. The two closest objects are then grouped, reducing the number of objects by one. The distance between this new "cluster" and the remaining objects is calculated. (B–F) The process repeats with the closest objects being fused into a new cluster until all objects are placed in a single cluster. (G) A dendrogram is then drawn representing the relationships between samples. It is important to note that dendrograms are not unique – there are many ways to represent the same relationships. The branch structure defines the relatedness of various samples. (I) An unordered data set, subjected to average linkage hierarchical clustering (J) reveals underlying patterns that can help identify classes in the data set.


A
S1
S6
S3
S7
S11
S5
S4
S9
S10
S2
S8
B
S1
S5
S2
1
S11
S3
S6
S7
2
S10
S9
S8
S4
3
C
2
1
3
D
S2
S4
S10
1
S1
S3
S6
S7
2
S5
S9
S8
S11
3
E
2
3
1
F
S10
S8
S2
1
S1
S3
S6
S7
2
S5
S9
S11
S4
3
G
H

5.3 Other Unsupervised Approaches

There are many approaches that have been applied to unsupervised analysis, including SOMs [85, 88, 95], self-organizing trees (SOTA) [36], relevance networks [18], force-directed layouts [47], principal component analysis [71] and others. Fundamentally, each of these uses some feature of the data and a rule for determining relationships to group genes (or samples) that share similar patterns of expression. In the context of disease analysis, all of these can be extremely useful for identifying new subclasses in the data – provided that the classes are reproducible and that they can be related to other clinical data. All of these algorithms will divide data into clusters, but whether the clusters are meaningful requires expert input and analysis. Critical assessment of the results is essential. There are anecdotal reports of clusters being found that separate data based on the hospital in which the sample was collected, the technician who ran the microarray assay or the day of the week on which the array was run. Clearly arrays can be very sensitive – one just has to filter the biological signal from the noise.

Chapter 27 addresses issue of data clustering in the context of elucidating gene function from expression data. Chapter 26 gives a detailed account of more advanced unsupervised methods for classifying expression data.

6 Classification

The goal in classification is to use supervised approaches to first find genes that separate samples into different clinical classes in the data, and then to implement an algorithm that can take the data from a new sample and, based on the patterns observed, assign that new sample to one of the previously identified classes.

One starts by assigning samples to particular biological classes based on some objective criteria. For example, in looking at leukemia samples, I may know for some initial set of data which patients have ALL and which have AML. The first question to be asked is: "Which genes best distinguish the various classes in the data?". The goal at this stage is to find those genes that

Figure 3 k-means clustering (A) begins with a collection of objects that (B) are randomly assigned to some number k of clusters, where k is specified by the user based on some intuition regarding the number of groups represented in the data. For each group, an average expression vector, represented by the hexagons, is calculated. (C) These average vectors are then used as "seeds" to form k new clusters, with objects reassigned to the new cluster whose average they are closest to. (D) Averages are computed for the new clusters, and the process repeats itself (E and F) until it converges with stable clusters. (G) An initial data set representing tumors from two organ sites (H) can be split by tissue type using k-means clustering with $k = 2$.

are most informative for distinguishing the samples based on class. Fortunately, there are a wide variety of statistical tools that can be brought to bear on this question, including t-tests (for two classes) and analysis of variance (ANOVA; for three or more classes) that assign p-values to genes based on their ability to distinguish between groups. One concern with these statistical approaches is the problem of multiple testing. Simply put, in an array with 10 000 genes, applying a 95% confidence limit on gene selection ($p \leq 0.05$) means that, by chance, one would expect to find 500 genes as significant. Clearly, we need to be more stringent in our gene selection. However, the important thing to remember is that what these methods provide are a means for prioritizing genes for further analysis. It should be noted that there are other widely used approaches, such as significance analysis of microarrays (SAM) [89], which uses an adjusted t-statistic (or F-statistic), modified to correct for overestimates arising from small values in the denominator, along with permutation testing to estimate the false discovery rate (FDR) in any selected significant gene set. Chapter 25 goes into detail about the statistics of selecting significantly differentially expressed genes.

The second key element is the selection of an appropriate classification algorithm. There are a wide range of algorithms that have been used for classification, including weighted voting [34], ANNs [14, 31], discriminant analysis [4,52,62,64], classification and regression trees (CART) [15], SVMs [17, 70], and k-nearest neighbors (kNN) [87], as well as a host of others. Essentially each of these uses an original set of samples, or training set, to develop a rule that takes a new test sample from a test set and uses its expression vector sample, trimmed to a previously identified set of classification genes, to place this test sample into the context of the original sample set, thus identifying its class.

6.1 kNN Classification

In many ways, kNN is the simplest approach to doing classification (Figure 4). First, we must assemble a collection of expression vectors for our samples and assign the samples to various experimental classes. We will refer to these samples, about which we have prior knowledge, as our *training set*. It is also useful to have a second collection of samples on which we will test the algorithm, known as the *test set*. Using the training set of samples, we then select genes that separate the various classes using an appropriate statistical test to identify good classification candidate genes, thus reducing the size of the sample classification vectors. This represents a first-pass collection of classification genes. The next step is to identify and eliminate samples that appear to be outliers. These may be important because they possibly represent new subclasses in our original sample classification set; alternatively, they

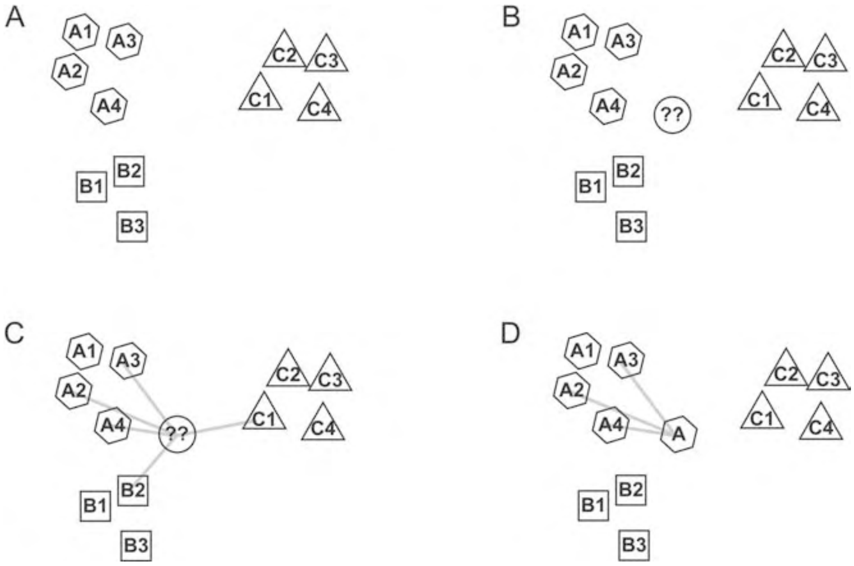


Figure 4 *k*NN classification (A) starts with a well-defined set of training samples that fall into distinct classes. (B) A sample to be classified is added from a test set. (C) The parameter *k* specifies the number of neighbors to be used in classification; for $k = 5$, the five nearest neighbors (based on some distance measure) are found. (D) Among the *k* nearest neighbors, majority rule determines the class of the test sample.

may just represent poor-quality data. The outlying samples are identified by applying a correlation filter to the reduced sample expression vectors:

(i) The Pearson correlation coefficient r is computed between a given vector and each member of the training set; the maximum r identified is called the $r_{\max}$ for that vector.

(ii) The vector is randomized a user-specified number of times, and each time, an $r_{\max}$ is calculated using the randomized vector (call it $r*_{\max}$), just as in step (i).

(iii) The fraction of times $r*_{\max}$ exceeds $r_{\max}$ over all randomizations is used to calculate a p-value for that vector.

(iv) If the p-value for a vector is less than a user-specified threshold (meaning it is well correlated with other samples), that vector is retained for further analysis; otherwise, it is discarded.

(v) Steps (i)–(iv) are repeated for every sample vector in the set.

At this stage, we have a collection of sample vectors, our training set, that represent our prior knowledge of the biological classes represented in the data.

We now turn our attention to the assigning new samples in our test set to classes based on their expression vectors and attempt to classify these test samples. For each new sample in the test set, we reduce its expression vector to include only those genes previously identified as being significant for classification. We then compute the distance between this reduced expression vector and the reduced expression vectors for each and every sample in the training set. As the name *k*NN implies, we choose some number k of nearest neighbors from the training set – those k vectors that have the smallest distances from our test sample. We then simply assign the new test vector to the class most highly represented in its k nearest neighbors. If there is a tie, the new sample is unclassified.

Chapter 26 goes into more detail about how to classify samples of expression data.

7 Validation

Ideally, to validate a classification method, it is most useful to have a set of samples in the test set that is independent of those used in the training set. In practice, microarray studies often have a limited number of samples and these are needed for building and training the algorithm. An alternative to using an independent test set is to do *leave*-k-*out cross-validation* (LKOCV) [81]. As one might guess, this approach leaves out some subset of the initial collection of N samples, develops a classifier using the $(N - k)$ samples that remain and applies it to k samples in the test set. This process is then repeated choosing a new set of k vectors to be left out and classified, and the process repeats itself. The simplest approach is to simply do leave-one-out cross-validation (LOOCV).

While this approach can be extremely useful when we lack an independent test set, it is often applied inappropriately as a partial rather than a full cross-validation. The distinction is the stage in the process where one "leaves k out". Many published studies have used their entire data set to select a set of classification genes, and *then* divide the samples into k and $(N - k)$ sample test and training sets. In fact, this has the potential to bias the results because the test and training sets are not independent as all of the samples were used to select the classification gene set. In particular, the presence of all of the samples in the initial gene selection process may favorably bias the ultimate success of any classifier that is constructed.

In *full* LKOCV, the data is divided into k and $(N - k)$ sample test and training sets, and the $(N - k)$ training set is used to select a classification gene set and then to apply it to creating a classification algorithm and using it to classifying the k test samples. One can then estimate the accuracy of the classification system by simply averaging over the complete set of classifiers.

8 Sample Selection and Classification

The choice of samples for training is an important but often neglected element of developing a classifier. One of the most important issues in sample selection is balancing representation of sample classes, as well as making sure that the analysis is not confounded by other factors. Nearly all algorithms work by a majority consensus rule and if we have two classes, A and B, with eight in class A and two in class B, the simplest classifier would simply assign everything to class A with 80% accuracy – a result that clearly would not be widely applicable. We also need to make sure that the samples are selected such that there are no confounding factors. For example, if we wanted to classify patients based on survival and all of the surviving patients received adjuvant chemotherapy, while the patients in the nonsurviving class did not, it is not likely that the patterns we find and use for classification will be applicable to a more heterogeneous population. It is also clear that selecting a sample of sufficient size to resolve classes is an important consideration [22, 61, 80]. One important aspect of this problem was recently illustrated in a publication by Radich and colleagues [69], which analyzed gene expression levels in peripheral blood and demonstrated significant but reproducible inter-individual variation in expression for a 1130 genes (selected from an array assaying 24 000 transcripts). What this suggests is that a small sample size *may* lead to biases in the gene selection set due to random effects in assigning patients to classes. These differences may be biological in nature or they may be due to systematic effects such as where the experimental population lives or the protocol used for sample collection. A truly independent test set would allow the results from a study to be validated in a much more meaningful and robust manner. Chapter 26 gives a more detailed statistical account of supervised methods for classifying expression data.

9 Limitations and Success of Classification

Although there have been attempts to identify "the best" classification approach, the evidence suggests that there is no single method that will work in all cases and, similarly, that many methods may work in any particular case. What is important is to understand the limitations of the approach. Most of the studies that have been conducted to date have involved relatively small numbers of patients and it is not at all clear how these results will generalize to larger clinical populations, with samples collected across a number of sites, with variations in sample collection and handling that occur outside of a well controlled laboratory-based study.

One potential difficulty in using expression profiles for classification is that very often the signatures that are identified are not easily interpreted causally or mechanistically with respect to the underlying disease. Ultimately, finding genes that can be functionally linked to outcome may provide insight into possible therapeutic interventions. However, the failure to provide a biological interpretation does not diminish the potential clinical utility of well-established biomarkers. It should be noted that there are many examples of biomarkers of unknown function, such as prostate-specific antigen (PSA) or carcinoembryonic antigen (CEA), that are extremely useful as diagnostic or prognostic markers for various diseases. It may be more useful to consider gene lists emerging from classification experiments as nothing more that sets of biomarkers with clinical applications; if they have a biological interpretation, this is simply a bonus.

10 Data Reporting and Comparisons

As noted previously, many microarray studies are underpowered in the sense that they do not include enough samples to draw firm conclusions without the collection of additional samples that can be used for validation. While collecting new samples, particularly patient samples that require significant follow-up, can be a challenge, the increasing number of published microarray studies provides an opportunity for the analysis of additional, independently derived data sets. To facilitate comparisons between studies, the Microarray Gene Expression Data (MGED) society developed a set of standards for data reporting known as MIAME (the Minimal Information About a Microarray Experiment) [7–10, 16].

The MIAME standards attempt to capture all of the information necessary to fully describe a particular microarray experiment. The driving principle in developing MIAME was an attempt to answer the question: “What would an independent scientist need to know to analyze a particular published experiment?”. In discussing this problem, the MGED group came to the realization that answering this question required nearly complete descriptions of all aspects of the experiment, including the composition and construction of the microarray platform itself (and specifically the sequence of the individual probes used on the arrays), the protocols used for RNA labeling and hybridization, the design of the hybridization assays performed (detailing, in two-color assays, which samples are compared on each array), the methods used for data extraction and analysis, and, most importantly, the design and implementation of the method used to assemble the samples including a detailed description of those samples.

Although there was some initial resistance to MIAME (as it required, what some termed, maximal rather than minimal information) the research community came to acknowledge that the requested was essential to effective data interpretation. At a very high level, it is the samples and their annotation that drive the analysis (e.g. what genes separate tumor and normal tissue). However, the genes selected in a study depend on understanding what genes are represented on the array and probe sequence allows the results to evolve as our understanding of the genome does, as well as providing a basis for comparison between studies. Microarrays are sensitive enough that we now know that minor variations in protocols, such as between hospitals or in creating different lots of arrays, can often be detected in the assays, and so even that level of detail is useful. Further, all of the results depend on the methods used for data analysis and different approaches can produce different results.

The major DNA sequence databases have developed gene expression data repositories that require MIAME-compliant data, and most journals now require that both raw and transformed data be submitted to one of these repositories; efforts are ongoing, both in the public and private sectors, to develop software that captures and presents data consistent with the MIAME standards. There is tremendous value in making gene expression data sets publicly available. In addition to serving as a source of independent data that can be used as a means of validating results, larger and more diverse sample populations, including cross-species comparisons, can provide more robust data sets for "meta-analysis" designed to find universal patterns of gene expression that can be associated with a given biological system [57, 84].

One must also exercise caution in comparing data sets between laboratories. Although there have been many successful applications of microarray analysis, often with high rates of validation using an alternate technology such as Northern analysis or quantitative reverse transcription polymerase chain reaction (qRT-PCR), a number of published studies have called into question the validity of microarray assays, in part because of observed disparities between results obtained by different groups analyzing similar samples [48, 55, 56, 65, 72, 79, 90, 101]. However, in many instances, it seems that the failure to find concordance between microarray platforms designed to assay biologically relevant patterns of expression is a failure not of the platform or the biological system, but rather a reflection of metrics used to evaluate concordance. Other meta-analyses focus on overlapping lists of significant genes, neglecting the fact that in many instances these are derived from not only different platforms, but also vastly different approaches to data analysis [42, 55, 86] – an effect can be seen even in looking at a single data set generated on a single platform. When this has been compensated for, the results generally show good concordance between different laboratories and the various array types [11, 19, 38, 44, 94, 103].

This problem of between-platform and -laboratory comparison was systematically dealt with in a series of papers that appeared in the May 2005 issue of *Nature Methods* [41, 51, 96]. Larkin and coworkers [51] analyzed gene expression in a mouse model of hypertension and compared results obtained using spotted cDNA arrays and Affymetrix GeneChips^{TM}. What they found was that for the genes that could be compared, 88% showed expression patterns that were driven by the underlying biology rather than the platform and that these genes also correlated well with real-time qRT-PCR). Surprisingly, the 12% of genes that showed platform-specific effects also correlated poorly with qRT-PCR. When comparing these platform discrepant genes to the platform concordant genes, it was found that the discrepant genes were much more likely to map to poorly annotated regions of the genome and consequently more likely to represent different splice forms. Irizarry and colleagues [41] compared gene expression using pairs of defined RNA samples and looked at a variety of platforms with data generated by a number of laboratories using a variety of microarray platforms. What Irizarry showed is that one can estimate the "lab effect," which encompasses differences in sites, platforms and protocols, and in doing so arrive at estimates of gene expression that can be compared between laboratories. Finally, the Toxicogenomics Research Consortium [96] reported that a careful standardization of laboratory and data analysis protocols resulted in a dramatic increase in concordance between the results obtained by different laboratories. The general conclusion, arrived at by these three groups independently, is that if experiments are done and analyzed carefully and systematically, the results are quite reproducible and provide insight to the underlying biology driving the systems being analyzed.

There are a number of other efforts to improve the overall utility of microarray data. MGED continues to develop standards for data reporting in standardized formats, particularly through the creation of the MGED ontology to describe experiments in a consistent fashion and an MGED working group is also seeking to develop objective quality standards for DNA microarrays. At present, this is seen as one of the greatest challenges in extending the utility of microarray analyses. In many ways, the development of objective quality scores for DNA sequence changed the way in which sequence data were used and the general consensus is that a similar transformation is necessary in the microarray field.

The challenge here, just as in data analysis, is that functional genomics data are much more complex than genome sequence data. One can define quality metrics on the level of the individual array probes, at the level of an entire array or in the context of a particular study comprised of hundreds of arrays. The real questions come down to understanding whether a particular assay is useful for the identification of genes that can be correlated with particular phenotypes and how high a level of confidence one can place in the results. There are various approaches that have been proposed ranging from intrinsic measures starting with signal-to-noise for each probe and analysis of replicates to the use of extrinsic standards, such as exogenous RNAs “spiked” into each assay. Likely, a combination of these approaches will emerge and prove to be useful.

Following the success of MIAME, a number of groups are attempting to extend MIAME to better capture information relevant to their particular disciplines (http://www.mged.org/Workgroups/rsbi/rsbi.html), including a MIAME/Tox for toxicogenomics, MIAME/Env for environmental exposures [75] and MIAME/Nut for nutrigenomics [33]. Work is also ongoing to develop standards for proteomics – the Minimal Information About a Proteomics Experiment (MIAPE [63]) – and for metabolomic/metabolonomic profiling – the Standard Metabolic Reporting Structure ([53]; http://www.smrsgroup.org). Finally, the External RNA Control Consortium (http://www.cstl.nist.gov/-biotech/workshops/ERCC2004/) [24] is attempting to create a set of well-defined RNA samples that can be used in as a control and calibration standard for a wide range of gene expression technologies.

11 Meta-analysis

Regardless of the initial goal of any microarray analysis, the most satisfying analysis yields insight into the underlying biological processes and this tends to be the most significant challenge that one must address in any study. The results of most analyses are a long list of genes that somehow need to be placed into a broader context. Further, additional data sources to the analysis can help provide constraints to mitigate the potential problems of multiple testing by providing independent pieces of evidence that the genes selected from a microarray study really do contribute to the underlying biology.

Fortunately, there is a long history of biological investigation and many tools and techniques that can be used effectively to further classify the data and aid in its interpretation. Some approaches build on relationships found by linking genes to PubMed abstracts or the associated Medical Subject Heading (MeSH) terms [26,28,32,43,58,104]. Others use constraints from the biological system under analysis such as using genetic linkage or quantitative trait locus (QTL) maps to narrow down the set of significant genes to those mapping to regions of the genome associated with appropriate trait [23, 27, 49, 77], although this can miss genes that contain causative mutations but which are not differentially regulated [23]. In solid tumor studies, one can analyze correlations between expression patterns and genome deletions or amplifications as determined by comparative genomic hybridization on arrays (array CGH) [21,35]. In developmental imprinting studies, gene expression may be compared to patterns of methylation [20,29], etc.

Another approach is to use the properties of the array itself to extract additional information. Ideally, each probe on the array should have been designed to query a specific transcript and each array element should carry annotation describing the gene and its known or putative function. There are many sources that can be drawn upon to provide classification that can be used for further analysis.

The Gene Ontology (GO) project (http://www.geneontology.org) attempts to classify gene products, assigning proteins to groups specifying their Molecular Function, the Biological Process to which they contribute, and their Cellular Component [6]. The GO terms in each class form a hierarchy of increasing specificity [formally, a directed acyclic graph (DAG)] so that the broadest classifications provide a general picture of the functional class to which a gene belongs (e.g. a kinase), while more precise terms will specify precisely what a particular gene does (such as specifying the substrate on which a kinase acts). As not all genes have a complete functional classification, increasing specificity reduces the sensitivity for placing genes into a particular functional class. As the functions of many genes have not been fully explored, there are some advantages to using the less-specific classes, which are often assigned based on sequence homology searches, for some analyses. Similarly, using Enzyme Commission (EC) numbers, genes can be mapped to metabolic and signaling pathway databases such as KEGG (Kyoto Encyclopedia of Genes and Genomes, http://www.genome.ad.jp/kegg) [45]. Mapping sequences to the genome sequence available for an increasing number of species can put the data into the context of genetic mapping results (e.g. QTL maps), chromosomal amplification and deletion data (important for the analysis of many forms of cancer), and allow analysis of epigenetic effects (such as promoter analysis from computational searches or chromatin immunoprecipitation analysis of transcription factor binding). Even linking array elements to the associated PubMed abstracts can provide insight given some final significant gene list.

Another very attractive approach is to use the properties of the data and the construction of the array to look for significant functional associations. You will recall that one of the key elements in establishing an array platform is the annotation of the arrayed probe elements. Imagine, for example, that 20% of the genes on the array are annotated as belonging to GO categories representing energy metabolism. If this is the case, then if we were to select, at random, a collection of "significant" genes would most likely have approximately 20% of its elements as belonging to the same energy metabolism class. In fact, we might not be very surprised if 30% of the genes in our significant set were energy metabolism genes; however, if the fraction were 80%, it might suggest that our experiment affected energy metabolism with a much higher frequency than one might expect by chance. Such insight may indeed provide clues as to the mechanisms at work in the biological system under study.

One widely used approach to leverage these classification schemes starts with the genes in a selected "significant" set that was selected based on the biology of phenotype under study. In looking at the classification assigned to these genes, as well as the classification assigned to the collection of genes on the array, one simply asks whether any particular class is overrepresented in the significant set and the Fisher exact test does precisely that. While doing this for any particular class is relatively straightforward, applying this method to a large collection assigned classes such as KEGG pathways or GO terms can be a huge challenge; for GO, what we really want to do is to consider all possible terms in the GO hierarchy. Fortunately, there are some software tools, including MAPPFinder [28], GOMiner [104] and EASE [37] (also implemented in MeV [74]) that calculate p-values for GO, KEGG, GenMAPP, Pfam and SMART protein domain assignments, promoter elements and a range of other classification systems.

The advantage of this approach is that it can provide a high-level view of functional classes or pathways that might be significantly affected in an experiment. However, we have to remember that these are functional classes, not functions, and deciphering these requires significant additional effort. It should also be noted that the only genes on each array that can be analyzed using this approach are those for which we have functional assignments. In any given experiment, one often finds a number of unknown expressed sequence tags (ESTs) and predicted genes that do not have defined functions and these drop out of any category-based meta-analysis. One approach to "rescue" these is to look for functional classes that pass a significance test and then to search the data for other genes that have similar patterns of expression across the sample. This provides a testable hypothesis that the genes of unknown function belong that that particular functional class.

If large, well-curated databases exist, such analysis can be carried out in interesting ways. For example Stuart and colleagues [84] started with gene expression data representing more than 3000 assays collected from a range of experiments in human, *Drosophila*, *Caenorhabditis elegans* and yeast, and stored in a common format in the Stanford Microarray Database (http://genome-www5.stanford.edu). By first identifying likely orthologs across all four species and then searching for orthologs that were highly correlated across the evolutionary history represented in the collection, they were able to test the hypothesis that genes fulfilling core biological functions should have conserved patterns of expression. Among their findings were a set of genes not previously associated with cell proliferation/cell cycle that the exhibit significant association with known genes mapped to these fundamental pathways. In order to test the hypothesis that the unknown genes were, in fact, involved in the cell cycle, Stuart and colleagues first looked at gene expression in a previously published human study comparing pancreatic tumors and normal tissue [39], and discovered that these genes were highly up-regulated in the rapidly dividing cancer cells. To further lend support to the association of these genes with cell cycle and proliferation functions, a RNA interference experiment was performed in *C. elegans* using one of these genes (ZK652.1) and the resulting loss of function phenotype included cells in the germline containing excess nuclei, suggesting the role of the gene is to suppress germline proliferation.

While significant process has been made in this area, our approaches to meta-analysis remain primitive. A great deal of work remains to be done in this area and as well-annotated, high-quality data sets are assembled and stored in databases such as GEO and ArrayExpress, our hope is that these will facilitate the creation of new, flexible software tools that can more effectively be used to discover function.

Part VIII of the book (Chapters 29–35) discusses in detail bioinformatics approaches towards elucidating gene and protein function, starting with approaches that reside upon a certain kind data, e.g. sequence, structure, free text, etc., and ending in a discussion of methods for integrating different kinds of data.

12 The Path Forward

For the time being, it appears that the greatest impact of arrays will remain their use in classification and, although these are still in there infancy, they are starting to see broader applications. One such example is The Netherlands Breast Cancer Study [91], which sought to distinguish between patients with the same stage of disease, but different response to treatment and overall outcome. The study was motivated by the observation that the best clinical predictors for metastasis, including lymph node status and histological grade, did not provide adequate prediction of clinical outcome. As a result many patients receive chemotherapy or hormonal therapy regardless of whether they need this additional treatment. The goal of their analysis was to identify signatures that would allow for individually tailored therapeutic strategies. By profiling tumors from 117 young patients and looking for correlations with clinical outcome, they were able to identify a “poor prognosis” signature comprised of 70 genes that was predictive of a short interval to distant metastasis in lymph node negative patients. Their analysis demonstrated that microarray-based signatures could outperform any clinically based predictions of outcome in identifying those patients who would benefit most from adjuvant therapy. The success of this initial study motivated a more extensive independent follow-up study involving 295 patients [92] that showed that the 70 gene classification profile was a more powerful predictor of the outcome of disease in young patients with breast cancer than standard systems based on clinical and histological criteria. The success of these two studies has led to a nationwide clinical trial in The Netherlands in which gene expression profiles for these 70 classifier genes are being collected on all breast cancer patients and used as an adjunct to classical clinical staging. Although we are still eagerly awaiting the outcome of this study, it is clear that the use of expression profiles as biomarkers to predict disease prognosis and outcome is coming of age.

References

1 International Human Genome Sequencing Consortium. 2004. Finishing the euchromatic sequence of the human genome. Nature **431**: 931–45.

2 Alizadeh, A., M. Eisen, R. Davis, et al. 2000. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. Nature **403**: 503–11.

3 Alon, U., N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack and A. J. Levine. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Natl Acad. Sci. USA **96**: 6745–50.

4 Antoniadis, A., S. Lambert-Lacroix and F. Leblanc. 2003. Effective dimension reduction methods for tumor classification using gene expression data. Bioinformatics **19**: 563–70.

5 Aronow, B. J., T. Toyokawa, A. Canning, K. Haghighi, U. Delling, E. Kranias, J. D. Molkentin and G. W. Dorn, 2nd. 2001. Divergent transcriptional responses to independent genetic causes of cardiac hypertrophy. Physiol. Genomics **6**: 19–28.

6 Ashburner, M., C. A. Ball, J. A. Blake, et al. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. **25**: 25–9.

7 Ball, C., A. Brazma, H. Causton, et al. 2004. Standards for microarray data: an open letter. Environ. Health. Perspect. **112**: A666.

8 Ball, C. A., A. Brazma, H. Causton, et al. 2004. Submission of microarray data to public repositories. PLoS Biol. **2**: e317.

9 Ball, C. A., G. Sherlock, H. Parkinson, et al. 2002. Standards for microarray data. Science **298**: 539.

10 Ball, C. A., G. Sherlock, H. Parkinson, et al. 2002. The underlying principles of scientific publication. Bioinformatics **18**: 1409.

11 Barczak, A., M. W. Rodriguez, K. Hanspers, L. L. Koth, Y. C. Tai, B. M. Bolstad, T. P. Speed and D. J. Erle. 2003. Spotted long oligonucleotide arrays for human gene expression analysis. Genome Res. **13**: 1775–85.

12 Beer, D. G., S. L. Kardia, C. C. Huang, et al. 2002. Gene-expression profiles predict survival of patients with lung adenocarcinoma. Nat. Med. **8**: 816–24.

13 Bhattacharjee, A., W. G. Richards, J. Staunton, et al. 2001. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. Proc. Natl Acad. Sci. USA **98**: 13790–5.

14 Bloom, G., I. V. Yang, D. Boulware, K. Y. Kwong, D. Coppola, S.

ESCHRICH, J. QUACKENBUSH AND T. J. YEATMAN. 2004. Multi-platform, multi-site, microarray-based human tumor classification. Am. J. Pathol. **164**: 9–16.

15 BOULESTEIX, A. L., G. TUTZ AND K. STRIMMER. 2003. A CART-based approach to discover emerging patterns in microarray data. Bioinformatics **19**: 2465–72.

16 BRAZMA, A., P. HINGAMP, J. QUACKENBUSH, et al. 2001. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. Nat. Genet. **29**: 365–71.

17 BROWN, M. P., W. N. GRUNDY, D. LIN, N. CRISTIANINI, C. W. SUGNET, T. S. FUREY, M. ARES, JR. AND D. HAUSSLER. 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc. Natl Acad. Sci. USA **97**: 262–7.

18 BUTTE, A. J. AND I. S. KOHANE. 1999. Unsupervised knowledge discovery in medical databases using relevance networks. Proc. AMIA Symp.: 711–5.

19 CARTER, M. G., T. HAMATANI, A. A. SHAROV, et al. 2003. *In situ*-synthesized novel microarray optimized for mouse stem cell and early developmental expression profiling. Genome Res. **13**: 1011–21.

20 CHEN, H., J. LIU, C. Q. ZHAO, B. A. DIWAN, B. A. MERRICK AND M. P. WAALKES. 2001. Association of c-*myc* overexpression and hyperproliferation with arsenite-induced malignant transformation. Toxicol. Appl. Pharmacol. **175**: 260–8.

21 CHEUNG, S. T., X. CHEN, X. Y. GUAN, S. Y. WONG, L. S. TAI, I. O. NG, S. SO AND S. T. FAN. 2002. Identify metastasis-associated genes in hepatocellular carcinoma through clonality delineation for multinodular tumor. Cancer Res. **62**: 4711–21.

22 CHURCHILL, G. A. 2002. Fundamentals of experimental design for cDNA microarrays. Nat. Genet. **32 (Suppl.)**: 490–5.

23 COOK, D. N., S. WANG, Y. WANG, et al. 2004. Genetic regulation of endotoxin-induced airway disease. Genomics **83**: 961–9.

24 CRONIN, M., K. GHOSH, F. SISTARE, J. QUACKENBUSH, V. VILKER AND C. O'CONNELL. 2004. Universal RNA reference materials for gene expression. Clin. Chem. **50**: 1464–71.

25 DERISI, J., L. PENLAND, P. O. BROWN, M. L. BITTNER, P. S. MELTZER, M. RAY, Y. CHEN, Y. A. SU AND J. M. TRENT. 1996. Use of a cDNA microarray to analyse gene expression patterns in human cancer. Nat. Genet. **14**: 457–60.

26 DJEBBARI, A., S. KARAMYCHEVA, E. HOWE AND J. QUACKENBUSH. 2005. MeSHer: identifying biological concepts in microarray assays based on PubMed. references and MESH terms. Bioinformatics **21**: 3324–6.

27 DOERGE, R. W. 2002. Mapping and analysis of quantitative trait loci in experimental populations. Nat. Rev. Genet. **3**: 43–52.

28 DONIGER, S. W., N. SALOMONIS, K. D. DAHLQUIST, K. VRANIZAN, S. C. LAWLOR AND B. R. CONKLIN. 2003. MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. Genome Biol. **4**: R7.

29 EHRLICH, M. 2002. DNA hypomethylation, cancer, the immunodeficiency, centromeric region instability, facial anomalies syndrome and chromosomal rearrangements. J. Nutr. **132**: 2424S–9S.

30 EISEN, M. B., P. T. SPELLMAN, P. O. BROWN AND D. BOTSTEIN. 1998. Cluster analysis and display of genome-wide expression patterns. Proc. Natl Acad. Sci. USA **95**: 14863–8.

31 ELLIS, M., N. DAVIS, A. COOP, et al. 2002. Development and validation of a method for using breast core needle biopsies for gene expression microarray analyses. Clin. Cancer Res. **8**: 1155–66.

32 FINK, J. L., S. DREWES, H. PATEL, J. B. WELSH, D. R. MASYS, J. CORBEIL AND M. GRIBSKOV. 2003. 2HAPI: a microarray data analysis system. Bioinformatics **19**: 1443–5.

33 GAROSI, P., C. DE FILIPPO, M. VAN ERK, P. ROCCA-SERRA, S. A. SANSONE AND R. ELLIOTT. 2005. Defining best practice for microarray analyses in nutrigenomic studies. Br. J. Nutr. **93**: 425–32.

34 GOLUB, T. R., D. K. SLONIM, P. TAMAYO, et al. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science **286**: 531–7.

35 GRAY, J. W. AND C. COLLINS. 2000. Genome changes and gene expression in human solid tumors. Carcinogenesis **21**: 443–52.

36 HERRERO, J., A. VALENCIA AND J. DOPAZO. 2001. A hierarchical unsupervised growing neural network for clustering gene expression patterns. Bioinformatics **17**: 126–36.

37 HOSACK, D. A., G. DENNIS, JR., B. T. SHERMAN, H. C. LANE AND R. A. LEMPICKI. 2003. Identifying biological themes within lists of genes with EASE. Genome Biol. **4**: R70.

38 HUGHES, T. R., M. MAO, A. R. JONES, et al. 2001. Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer. Nat. Biotechnol. **19**: 342–7.

39 IACOBUZIO-DONAHUE, C. A., A. MAITRA, M. OLSEN, et al. 2003. Exploration of global gene expression patterns in pancreatic adenocarcinoma using cDNA microarrays. Am. J. Pathol. **162**: 1151–62.

40 IRIZARRY, R. A., B. M. BOLSTAD, F. COLLIN, L. M. COPE, B. HOBBS AND T. P. SPEED. 2003. Summaries of Affymetrix GeneChip probe level data. Nucleic Acids Res. **31**: e15.

41 IRIZARRY, R. A., D. WARREN, F. SPENCER, et al. 2004. Multiple-laboratory comparison of microarray platforms. Nat. Methods **2**: 345–50.

42 JARVINEN, A. K., S. HAUTANIEMI, H. EDGREN, P. AUVINEN, J. SAARELA, O. P. KALLIONIEMI AND O. MONNI. 2004. Are data from different gene expression microarray platforms comparable? Genomics **83**: 1164–8.

43 JENSSEN, T. K., A. LAEGREID, J. KOMOROWSKI AND E. HOVIG. 2001. A literature network of human genes for high-throughput analysis of gene expression. Nat. Genet. **28**: 21–8.

44 KANE, M. D., T. A. JATKOE, C. R. STUMPF, J. LU, J. D. THOMAS AND S. J. MADORE. 2000. Assessment of the sensitivity and specificity of oligonucleotide (50mer) microarrays. Nucleic Acids Res. **28**: 4552–7.

45 KANEHISA, M. 2002. The KEGG database. Novartis Found. Symp. **247**: 91–101; discussion 101–3, 119–28, 244–52.

46 KHAN, J., R. SIMON, M. BITTNER, et al. 1998. Gene expression profiling of alveolar rhabdomyosarcoma with cDNA microarrays. Cancer Res. **58**: 5009–13.

47 KIM, S. K., J. LUND, M. KIRALY, et al. 2001. A gene expression map for *Caenorhabditis elegans*. Science **293**: 2087–92.

48 KUO, W. P., T. K. JENSSEN, A. J. BUTTE, L. OHNO-MACHADO AND I. S. KOHANE. 2002. Analysis of matched mRNA measurements from two different microarray technologies. Bioinformatics **18**: 405–12.

49 KWITEK-BLACK, A. E. AND H. J. JACOB. 2001. The use of designer rats in the genetic dissection of hypertension. Curr. Hypertens. Rep. **3**: 12–8.

50 LANDER, E. S., L. M. LINTON, B. BIRREN, et al. 2001. Initial sequencing and analysis of the human genome. Nature **409**: 860–921.

51 LARKIN, J. E., B. C. FRANK, H. GAVRAS, R. SULTANA AND J. QUACKENBUSH. 2005. Independence and reproducibility across microarray platforms. Nat. Methods **2**: 337–44.

52 LE, Q. T., P. D. SUTPHIN, S. RAYCHAUDHURI, et al. 2003. Identification of osteopontin as a prognostic plasma marker for head and neck squamous cell carcinomas. Clin. Cancer Res. **9**: 59–67.

53 LINDON, J. C., J. K. NICHOLSON, E. HOLMES, et al. 2005. Summary recommendations for standardization and reporting of metabolic analyses. Nat. Biotechnol. **23**: 833–8.

54 Lipshutz, R. J., D. Morris, M. Chee, et al. 1995. Using oligonucleotide probe arrays to access genetic diversity. Biotechniques **19**: 442–7.

55 Mah, N., A. Thelin, T. Lu, S. Nikolaus, et al. 2004. A comparison of oligonucleotide and cDNA-based microarray systems. Physiol. Genomics **16**: 361–70.

56 Maitra, A., N. V. Adsay, P. Argani, C. Iacobuzio-Donahue, A. De Marzo, J. L. Cameron, C. J. Yeo and R. H. Hruban. 2003. Multicomponent analysis of the pancreatic adenocarcinoma progression model using a pancreatic intraepithelial neoplasia tissue microarray. Mod. Pathol. **16**: 902–12.

57 Malek, R. L., R. B. Irby, Q. M. Guo, et al. 2002. Identification of Src transformation fingerprint in human colon cancer. Oncogene **21**: 7256–65.

58 Masys, D. R., J. B. Welsh, J. Lynn Fink, M. Gribskov, I. Klacansky and J. Corbeil. 2001. Use of keyword hierarchies to interpret gene expression patterns. Bioinformatics **17**: 319–26.

59 Michaels, G. S., D. B. Carr, M. Askenazi, S. Fuhrman, X. Wen and R. Somogyi. 1998. Cluster analysis and data visualization of large-scale gene expression data. Pac. Symp. Biocomput.: 42–53.

60 Moch, H., P. Schraml, L. Bubendorf, et al. 1999. [Identification of prognostic parameters for renal cell carcinoma by cDNA arrays and cell chips]. Verh. Dtsch. Ges. Pathol. **83**: 225–32.

61 Mukherjee, S., P. Tamayo, S. Rogers, R. Rifkin, A. Engle, C. Campbell, T. R. Golub and J. P. Mesirov. 2003. Estimating dataset size requirements for classifying DNA microarray data. J. Comput. Biol. **10**: 119–42.

62 Nguyen, D. V. and D. M. Rocke. 2002. Tumor classification by partial least squares using microarray gene expression data. Bioinformatics **18**: 39–50.

63 Orchard, S., H. Hermjakob, R. K. Julian, Jr., K. Runte, D. Sherman, J. Wojcik, W. Zhu and R. Apweiler. 2004. Common interchange standards for proteomics data: public availability of tools and schema. Proteomics **4**: 490–1.

64 Orr, M. S. and U. Scherf. 2002. Large-scale gene expression analysis in molecular target discovery. Leukemia **16**: 473–7.

65 Park, P. J., Y. A. Cao, S. Y. Lee, J. W. Kim, M. S. Chang, R. Hart and S. Choi. 2004. Current issues for DNA microarrays: platform comparison, double linear amplification, and universal RNA reference. J. Biotechnol. **112**: 225–45.

66 Perou, C. M., S. S. Jeffrey, M. van de Rijn, et al. 1999. Distinctive gene expression patterns in human mammary epithelial cells and breast cancers. Proc. Natl Acad. Sci. USA **96**: 9212–7.

67 Pomeroy, S. L., P. Tamayo, M. Gaasenbeek, et al. 2002. Prediction of central nervous system embryonal tumour outcome based on gene expression. Nature **415**: 436–42.

68 Quackenbush, J. 2002. Microarray data normalization and transformation. Nat. Genet. **32 (Suppl.)**: 496–501.

69 Radich, J. P., M. Mao, S. Stepaniants, et al. 2004. Individual-specific variation of gene expression in peripheral blood leukocytes. Genomics **83**: 980–8.

70 Ramaswamy, S., P. Tamayo, R. Rifkin, et al. 2001. Multiclass cancer diagnosis using tumor gene expression signatures. Proc. Natl Acad. Sci. USA **98**: 15149–54.

71 Raychaudhuri, S., J. M. Stuart and R. B. Altman. 2000. Principal components analysis to summarize microarray experiments: application to sporulation time series. Pac Symp Biocomput: 455–66.

72 Rogojina, A. T., W. E. Orr, B. K. Song and E. E. Geisert, Jr. 2003. Comparing the use of Affymetrix to spotted oligonucleotide microarrays using two retinal pigment epithelium cell lines. Mol. Vis. **9**: 482–96.

73 Ross, D. T., U. Scherf, M. B. Eisen, et al. 2000. Systematic variation in gene expression patterns in human cancer cell lines. Nat. Genet. **24**: 227–35.

74 Saeed, A. I., V. Sharov, J. White, et al. 2003. TM4: a free, open-source system for microarray data management and analysis. Biotechniques **34**: 374–8.

75 SANSONE, S., N. MORRISON, P. ROCCA-SERRA AND J. FOSTEL. 2005. Standardization initiatives in the (eco)toxicogenomics domain: a review. Comp. Funct. Genomics **8**: 633–41.

76 SCHADT, E. E., C. LI, B. ELLIS AND W. H. WONG. 2001. Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. J. Cell. Biochem. **Suppl. 37**: 120–5.

77 SCHADT, E. E., S. A. MONKS, T. A. DRAKE, et al. 2003. Genetics of gene expression surveyed in maize, mouse and man. Nature **422**: 297–302.

78 SCHENA, M., D. SHALON, R. W. DAVIS AND P. O. BROWN. 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science **270**: 467–70.

79 SHIPPY, R., T. J. SENDERA, R. LOCKNER, C. PALANIAPPAN, T. KAYSSER-KRANICH, G. WATTS AND J. ALSOBROOK. 2004. Performance evaluation of commercial short-oligonucleotide microarrays and the impact of noise in making cross-platform correlations. BMC Genomics **5**: 61.

80 SIMON, R., M. D. RADMACHER AND K. DOBBIN. 2002. Design of studies using DNA microarrays. Genet. Epidemiol. **23**: 21–36.

81 SIMON, R., M. D. RADMACHER, K. DOBBIN AND L. M. MCSHANE. 2003. Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. J. Natl Cancer Inst. **95**: 14–8.

82 SORLIE, T., C. M. PEROU, R. TIBSHIRANI, et al. 2001. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. Proc. Natl Acad. Sci. USA **98**: 10869–74.

83 SOUKAS, A., P. COHEN, N. D. SOCCI AND J. M. FRIEDMAN. 2000. Leptin-specific patterns of gene expression in white adipose tissue. Genes Dev. **14**: 963–80.

84 STUART, J. M., E. SEGAL, D. KOLLER AND S. K. KIM. 2003. A gene-coexpression network for global discovery of conserved genetic modules. Science **302**: 249–55.

85 TAMAYO, P., D. SLONIM, J. MESIROV, Q. ZHU, S. KITAREEWAN, E. DMITROVSKY, E. S. LANDER AND T. R. GOLUB. 1999. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc. Natl Acad. Sci. USA **96**: 2907–12.

86 TAN, P. K., T. J. DOWNEY, E. L. SPITZNAGEL, JR., et al. 2003. Evaluation of gene expression measurements from commercial microarray platforms. Nucleic Acids Res. **31**: 5676–84.

87 THEILHABER, J., T. CONNOLLY, S. ROMAN-ROMAN, S. BUSHNELL, A. JACKSON, K. CALL, T. GARCIA AND R. BARON. 2002. Finding genes in the C2C12 osteogenic pathway by *k*-nearest-neighbor classification of expression data. Genome Res. **12**: 165–76.

88 TORONEN, P., M. KOLEHMAINEN, G. WONG AND E. CASTREN. 1999. Analysis of gene expression data using self-organizing maps. FEBS Lett. **451**: 142–6.

89 TUSHER, V. G., R. TIBSHIRANI AND G. CHU. 2001. Significance analysis of microarrays applied to the ionizing radiation response. Proc. Natl Acad. Sci. USA **98**: 5116–21.

90 ULRICH, R. G., J. C. ROCKETT, G. G. GIBSON AND S. D. PETTIT. 2004. Overview of an interlaboratory collaboration on evaluating the effects of model hepatotoxicants on hepatic gene expression. Environ. Health Perspect. **112**: 423–7.

91 VAN 'T VEER, L. J., H. DAI, M. J. VAN DE VIJVER, et al. 2002. Gene expression profiling predicts clinical outcome of breast cancer. Nature **415**: 530–6.

92 VAN DE VIJVER, M. J., Y. D. HE, L. J. VAN 'T VEER, et al. 2002. A gene-expression signature as a predictor of survival in breast cancer. N. Engl. J. Med. **347**: 1999–2009.

93 VENTER, J. C., M. D. ADAMS, E. W. MYERS, et al. 2001. The sequence of the human genome. Science **291**: 1304–51.

94 WANG, H. T., J. P. KONG, F. DING, X. Q. WANG, M. R. WANG, L. X. LIU, M. WU AND Z. H. LIU. 2003. Analysis of gene expression profile induced by EMP-1 in esophageal cancer cells using cDNA microarray. World J. Gastroenterol. **9**: 392–8.

95 WANG, J., J. DELABIE, H. AASHEIM, E. SMELAND AND O. MYKLEBOST. 2002. Clustering of the SOM easily reveals distinct gene expression patterns: results of a reanalysis of lymphoma study. BMC Bioinformatics **3**: 36.

96 WEIS, B. K. 2005. Standardizing global gene expression analysis between laboratories and across platforms. Nat. Methods **2**: 351–6.

97 WELFORD, S. M., J. GREGG, E. CHEN, D. GARRISON, P. H. SORENSEN, C. T. DENNY AND S. F. NELSON. 1998. Detection of differentially expressed genes in primary tumor tissues using representational differences analysis coupled to microarray hybridization. Nucleic Acids Res. **26**: 3059–65.

98 WEN, X., S. FUHRMAN, G. S. MICHAELS, D. B. CARR, S. SMITH, J. L. BARKER AND R. SOMOGYI. 1998. Large-scale temporal gene expression mapping of central nervous system development. Proc. Natl Acad. Sci. USA **95**: 334–9.

99 YANG, I. V., E. CHEN, J. P. HASSEMAN, et al. 2002. Within the fold: assessing differential expression measures and reproducibility in microarray assays. Genome Biol. **3**: R62.

100 YANG, Y. H., S. DUDOIT, P. LUU, D. M. LIN, V. PENG, J. NGAI AND T. P. SPEED. 2002. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. Nucleic Acids Res. **30**: e15.

101 YAUK, C. L., M. L. BERNDT, A. WILLIAMS AND G. R. DOUGLAS. 2004. Comprehensive comparison of six microarray technologies. Nucleic Acids Res. **32**: e124.

102 YEUNG, K. Y., D. R. HAYNOR AND W. L. RUZZO. 2001. Validating clustering for gene expression data. Bioinformatics **17**: 309–18.

103 YUEN, T., E. WURMBACH, R. L. PFEFFER, B. J. EBERSOLE AND S. C. SEALFON. 2002. Accuracy and calibration of commercial oligonucleotide and custom cDNA microarrays. Nucleic Acids Res. **30**: e48.

104 ZEEBERG, B. R., W. FENG, G. WANG, et al. 2003. GoMiner: a resource for biological interpretation of genomic and proteomic data. Genome Biol. **4**: R28.

25
Low-level Analysis of Microarray Experiments

Wolfgang Huber, Anja von Heydebreck and Martin Vingron

1 Introduction

This chapter gives an overview over the methods used in the low-level analysis of gene expression data generated using DNA microarrays. This type of experiment allows to determine relative levels of nucleic acid abundance in a set of tissues or cell populations for thousands of transcripts or loci simultaneously. Careful statistical design and analysis are essential to improve the efficiency and reliability of microarray experiments throughout the data acquisition and analysis process. This includes the design of probes, experimental design, image analysis of microarray scanned images, normalization of fluorescence intensities, the assessment of the quality of microarray data and incorporation of quality information in subsequent analyses, combination of information across arrays and across sets of experiments, the discovery and recognition of patterns in expression at the single-gene and multiple-gene levels, and assessment of significance of these findings, considering the fact that there is a lot of noise and thus random features in the data. For all of these components, access to a flexible and efficient statistical computing environment is an essential aspect.

1.1 Microarray technology

In the context of the Human Genome Project, new technologies emerged that facilitate the parallel execution of experiments on a large number of genes simultaneously. The so-called *DNA microarrays*, or *DNA chips*, constitute a prominent example. This technology aims at the measurement of nucleic acid levels in particular cells or tissues for many genes or loci at once. Nucleic acids of interest can be polyadenylated RNA, total RNA or DNA. We will in the following use the term *gene* loosely to denote any unit of nucleic acid of interest. Single strands of complementary DNA for the genes to be considered are immobilized on spots arranged in a grid (*array*) on a support which will typically be a glass slide or a quartz wafer. The number of spots can range

Bioinformatics – From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

from dozens to millions. From a sample of interest, e.g. a tumor biopsy, the nucleic acid is extracted, labeled and hybridized to the array. Measuring the amount of label on each spot then yields an intensity measurement that should be correlated to the abundance of the corresponding gene in the sample. Chapter 24 goes into more detail regarding the experimental technology, therefore we only give a short summary here.

Two schemes of fluorescent labeling are in common use today. One variant labels a single sample. For example, the Affymetrix synthesizes sets of short oligomers on a glass wafer and uses a single fluorescent label ([26], see also www.affymetrix.com). Alternatively, two samples are labeled with a green and a red fluorescent dye, respectively. The mixture of the two nucleic acid preparations is then hybridized simultaneously to a common array on a glass slide. In the case where the probes are polymerase chain reaction (PCR) products from cDNA clones that are spotted on the array, this technology is usually refered to as the Stanford technology [12]. On the other hand, companies like Agilent have immobilized long oligomers of 60–70 bp length and used two-color labeling. The hybridization is quantified by a laser scanner that determines the intensities of each of the two labels over the entire array.

The parallelism in microarray experiments lies in the hybridization of nucleic acids extracted from a single sample to many genes simultaneously. The measured abundances, though, are usually not obtained on an absolute scale. This is because they depend on many hard-to-control factors such as the efficiencies of the various chemical reactions involved in the sample preparation, as well as on the amount of immobilized DNA available for hybridization.

Traditionally, one or a few probes were selected for each gene, based on known information on its sequence and structure. More recently, it has become possible to produce probes for the complete sequence content of a whole genome or for significant parts of it [3,6,32].

1.2 Prerequisites

A number of steps are involved in the generation of the raw data. The *experimental design* includes choice and collection of samples (tissue biopsies or cell lines exposed to different treatments), choice of probes and array platform, choice of controls, RNA extraction, amplification, labeling and hybridization procedures, allocation of replicates, and scheduling of the experiments. Careful planning is needed, as the quality of the experimental design determines to a large extent the utility of the data [7,23,42]. A fundamental guideline is the avoidance of *confounding* between different biological factors of interest or between a biological factor of interest and a technical factor that is anticipated to affect the measurements.

There are many different ways for the outline of a microarray experiment. In many cases, a development in time is studied leading to a series of hybridizations following each other. In a cohort study, different conditions like healthy/diseased or different disease types may be studied. In designed factorial experiments, one or several factors, e.g. treatment with a drug, genetic background and/or tissue type, are varied in a controlled manner. We generally refer to a time point or a state as a condition and typically for each condition several replicate hybridizations are performed. The replicates should provide the information necessary to judge the significance of the conclusions one wishes to draw from the comparison of the different conditions. When going deeper into the subject it soon becomes clear that this simple outline constitutes a challenging program.

1.3 Preprocessing

Preprocessing is the link between the raw experiment data and the higher-level statistical analysis. The five tasks of preprocessing can be summarized as follows: data import, background adjustment, normalization, summarization of multiple probes per transcript and quality control. They are driven by the properties of microarray technology. The data come in different formats and are often scattered across a number of files (or, possibly, database tables), from which they need to be extracted and unified. Part of the hybridization is nonspecific and the measured intensities are affected by noise in the optical detection. Therefore, the observed intensities need to be adjusted to give accurate measurements of specific hybridization. We refer to this aspect of preprocessing as *background adjustment*. Different efficiencies of reverse transcription, labeling or hybridization reactions among different arrays cause systematic technical biases and need to be corrected. We call the task of manipulating data to make measurements from different arrays comparable *normalization*. On some platforms, genes are represented with more than one probe. *Summarizing* the data is necessary when we want to reduce the measurements from various probes into one quantity that estimates the amount of RNA transcript. The reproducibility of measurements is limited by random fluctuations or measurement error. Basically, we can distinguish between two types of fluctuations: (i) those that affect individual measurements, and follow a localized distribution, and (ii) those, that affect whole groups of measurements, and are often drastic, large and irregular. The former type can be described with *error models*, while the latter is best dealt with by *quality control* procedures that try to detect and eliminate the affected measurements.

2 Visualization and Exploration of the Raw Data

A microarray experiment consists of the following components: a set of *probes*, an *array* on which these probes are immobilized at specified locations, a *sample* containing a complex mixture of labeled biomolecules that can bind to the probes and a *detector* that is able to measure the spatially resolved distribution of label after it has bound to the array. The probes are chosen such that they bind to specific sample molecules; for DNA arrays, this is ensured by the high sequence specificity of the hybridization reaction between complementary DNA strands. The array is typically a glass slide or a quartz wafer. The sample molecules are labeled through fluorescent dyes such as phycoerythrin, Cy3 or Cy5. After exposure of the array to the sample, the abundance of individual species of sample molecules can be quantified through the signal intensity at the matching probe sites. To facilitate direct comparison, the spotted array technology developed in Stanford [12] involves the simultaneous hybridization of two samples labeled with different fluorescent dyes, and detection at the two corresponding wavelengths. Figure 1 shows an example.

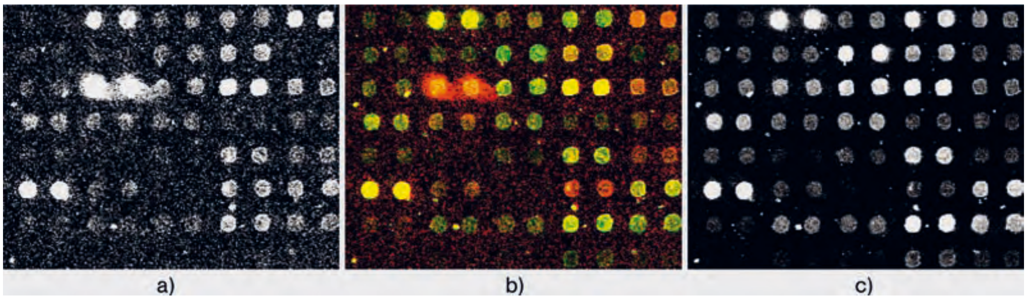


Figure 1 The detected intensity distributions from a cDNA microarray for a region comprising 40 probes spotted in duplicate. The total number of probes on an array may range from a few dozens to tens of thousands. (a) Grey-scale representation of the detected label fluorescence at 635 nm (red), corresponding to mRNA sample A. (c) Label fluorescence at 532 nm (green), corresponding to mRNA sample B. (b) False-color overlay image from the two intensity distributions. The spots are red, green or yellow, depending on whether the gene is transcribed only in sample A, sample B or both.

2.1 Image Analysis

In the *image analysis* step we extract probe intensities out of the scanned images, such as shown in Figures 1 and 2. The images are scanned by the detector at a high spatial resolution, such that each probe is represented by many pixels. In order to obtain a single overall intensity value for each probe, the corresponding pixels need to be identified (segmentation) and the intensities need to be summarized (quantification). In addition to the overall

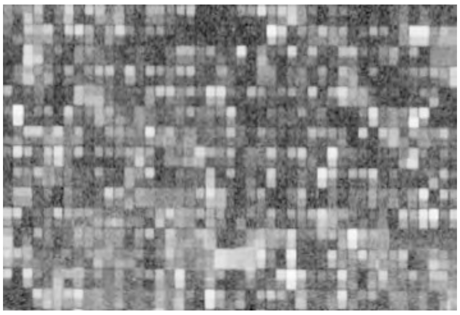


Figure 2 Gray-scale representation of the intensity distribution from a small sector of an Affymetrix HG-U133A genechip.

probe intensity, further auxiliary quantities may be calculated, such as an estimate of apparent unspecific "local background" intensity or spot quality measures.

Various software packages offer a variety of segmentation and quantification methods. They differ in their robustness against irregularities and in the amount of human interaction that they require. Different types of irregularities may occur in different types of microarray technology and a segmentation or quantification algorithm that is good for one platform is not necessarily suitable for another. For instance, the variation of spot shapes and positions that the segmentation has to deal with depends on the properties of the support and how the probes were attached to it (e.g. quill-pen type printing of PCR product, *in situ* oligonucleotide synthesis by ink jetting, *in situ* synthesis by photolithography). Furthermore, larger variations in the spot positioning from array to array can be expected in home-made arrays than in mass produced ones. An evaluation of image analysis methods for spotted cDNA arrays is described by Yang and coworkers [40].

For a microarray project, the image quantification marks the transition in the work flow from "wet-lab" procedures to computational ones. Hence, this is a good point to spend some effort looking at the quality and plausibility of the data. This has several aspects: confirm that positive and negative controls behave as expected, verify that replicates yield measurements close to each other, and check for the occurrence of artifacts, biases or errors. In the following we present a number of data exploration and visualization methods that may be useful for these tasks.

2.2 Dynamic Range and Spatial Effects

A simple and fundamental property of the data is the dynamic range and the distribution of intensities. Since many experimental problems occur at the level of a whole array or the sample preparation, it is instructive to look

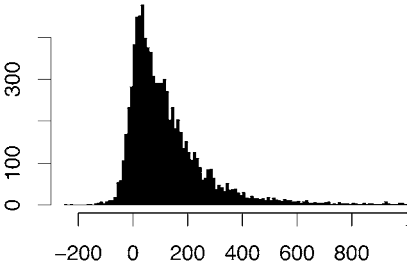


Figure 3 Histogram of probe intensities at the green wavelength for a cDNA microarray similar to the one depicted in Figure 1. The intensities were determined, in arbitrary units, by an image quantification method and "local background" intensities were subtracted. Due to measurement noise, this lead to nonpositive probe intensities for part of the genes with low or zero abundance. The x-axis has been cut off at the 99% quantile of the distribution. The maximum value is about 4000.

at the histogram of intensities from each sample. An example is shown in Figure 3. Typically, for arrays that contain quasi-random gene selections, one observes a unimodal distribution with most of its mass at small intensities, corresponding to genes that are not or only weakly transcribed in the sample, and a long tail to the right, corresponding to genes that are transcribed at various levels. In most cases, the occurence of multiple peaks in the histogram indicates an experimental artifact. To get an overview over multiple arrays, it is instructive to look at the box plots of the intensities from each sample. Problematic arrays should be excluded from further analysis.

Crude artifacts, such as scratches or spatial inhomogeneities, will usually be noticed already from the scanner image at the stage of the image quantification. Nevertheless, to get a quick and potentially more sensitive view of spatial effects, a false-color representation of the probe intensities as a function of their spatial coordinates can be useful. There are different options for the intensity scaling, among them the linear, logarithmic and rank scales. Each one will highlight different features of the spatial distribution. Examples are shown in Figure 4. A more sophisticated and more sensitive method to detect subtle artifacts is to look at the residuals of a probe-level model fitted for a set of arrays instead of the probe intensities themselves [5].

2.3 Scatterplot

Usually, the samples hybridized to a series of arrays are biologically related, such that the transcription levels of a large fraction of genes are approximately the same across the samples. This can be expected, for example, for cell cultures exposed to different conditions or for cells from biopsies of the same tissue type, possibly subject to different disease conditions. Visually, this can

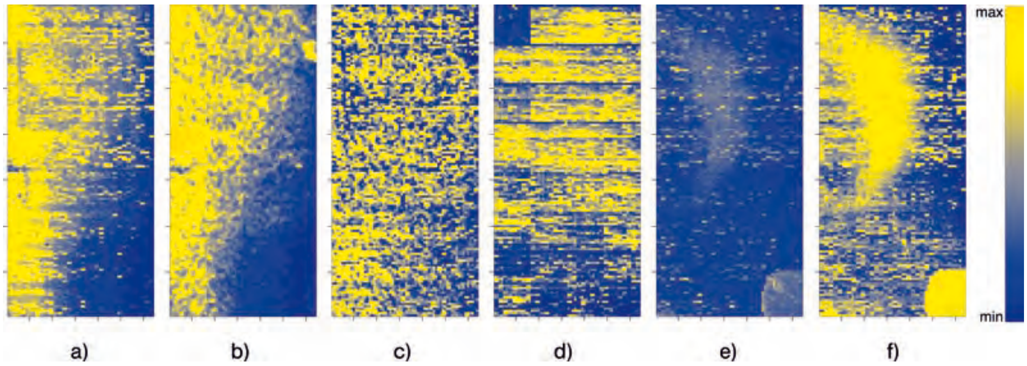


Figure 4 False-color representations of the spatial intensity distributions from three different 64×136 spot cDNA microarrays from one experiment series. The color scale is shown in the panel on the right. (a) Probe intensities in the red color channel, (b) local background intensities and (c) background-subtracted probe intensities. In (a) and (b), there is an artifactual intensity gradient, which is mostly removed in (c). For visualization, the color scale was chosen in each image to be proportional to the ranks of the intensities. (d) For a second array, probe intensities in the green color channel. There is a rectangular region of low intensity in the top left corner, corresponding to one print-tip. Apparently, there was a sporadic failure of the tip for this particular array. Panels (e) and (f) show the probe intensities in the green color channel from a third array. The color scale was chosen proportional to the logarithms of intensities in (e) and proportional to the ranks in (f). Here, the latter provides better contrast. Interestingly, the bright blob in the lower right corner appears only in the green color channel, while the half-moon-shaped region appears both in green and red (not shown).

be examined from the scatterplot of the probe intensities for a pair of samples. An example is shown in Figure 5.

The scatterplot allows us to assess both measurement noise and systematic biases. Ideally, the data from the majority of the genes that are unchanged should lie on the bisector of the scatterplot. In reality, there are both systematic and random deviations from this [33]. For instance, if the label incorporation rate and photoefficiency of the red dye were systematically lower than that of the green dye by a factor of 0.75, the data would be expected not to lie on the bisector, but rather on the line $y = 0.75x$.

Most of the data in Figure 5 is squeezed into a tiny corner in the bottom left of the plot. More informative displays may be obtained from other axis scalings. A frequently used choice is the double-logarithmic scale. An example is shown in Figure 6. It is customary to transform to new variables $A = (\log R + \log G)/2$, $M = \log R - \log G$ [11]. Up to a scale factor of $\sqrt{2}$, this corresponds to a coordinate system rotation by 45°. The horizontal coordinate A is a measure of average transcription level, while the *log-ratio* M is a measure for differential transcription. If the majority of genes are not differentially transcribed, the scatter of the data in the vertical direction may be considered a measure of the random variation. Figure 6(a) also shows a systematic

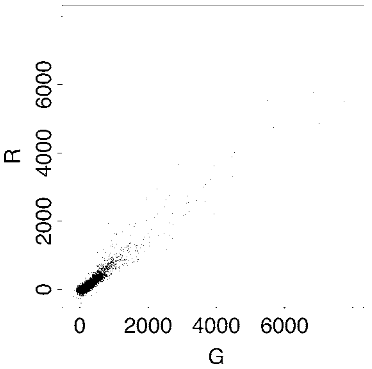


Figure 5 Scatterplot of probe intensities in the red and the green color channel from a cDNA array containing 8000 probes.

deviation of the observed values of M from the line $M = 0$, estimated through a local regression line (we used loess [8] with default parameters span = 0.75, degree = 2). There is an apparent dependence $M_0(A)$ of this deviation on the mean intensity A. However, this is most likely an artifact of applying the logarithmic transformation: as shown in Figure 6(b), the regression line may be modeled sufficiently well by a constant $M_0(A) = M_0$ if an appropriate offset is added to the R values before taking the logarithm. Note that a horizontal line at $M = M_0$ in Figure 6(b) corresponds to a straight line of slope 2^{M_0} and with intercept c in Figure 5.

Figure 6 shows the *heteroskedasticity* of log-ratios: while the variance of M is relatively small and approximately constant for large average intensities A, it becomes larger as A decreases. Conversely, examination of the differences $R - G$, e.g. through plots like in Figure 5, shows that their variance is smallest for small values of the average intensity $R + G$ and increases with $R + G$. Sometimes, one wishes to visualize the data in a manner such that the variance is constant along the whole dynamic range. A data transformation that achieves this goal is called a variance-stabilizing transformation. In fact, *homoskedastic* representations of the data are not only useful for visualization, but also for further statistical analyses. This will be discussed in more detail in Section 5.2.

Two extensions of the scatterplot are shown in Figures 7 and 8. Rather than plotting a symbol for every data point, they use a density representation, which may be useful for larger arrays. For example, Figure 7 shows the scatterplot from the comparison of two tissue samples based on 152 000 probes [the arrays used were RZPD Unigene-II arrays (www.rzpd.de)]. The point density in the central region of the plot is estimated by a kernel density estimator. Three-way comparisons may be performed through a projection such as in Figure 8. This uses the fact that the $(1, 1, 1)$-component of a three-way

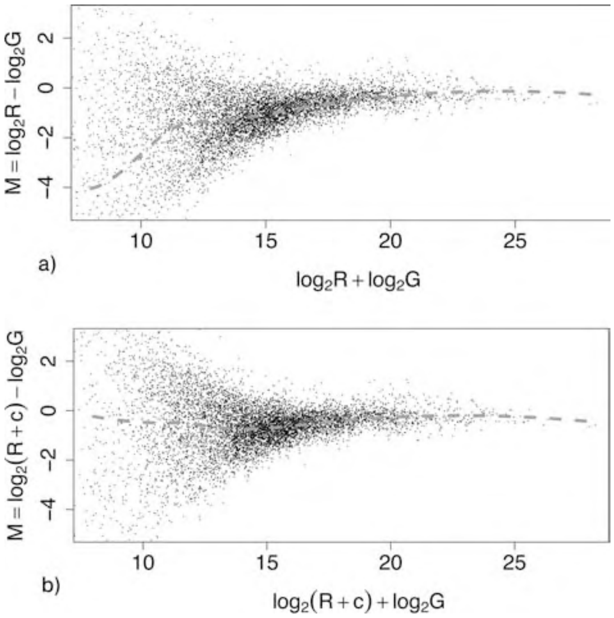


Figure 6 (a) The same data as in Figure 5, after logarithmic transformation and clockwise rotation by 45°. The dashed line shows a local regression estimate of the systematic effect $M_0(A)$, see text. (b) similar to (a); however, a constant value $c = 42$ has been added to the red intensities before log transformation. After this, the estimated curve for the systematic effect $M_0(A)$ is approximately constant.

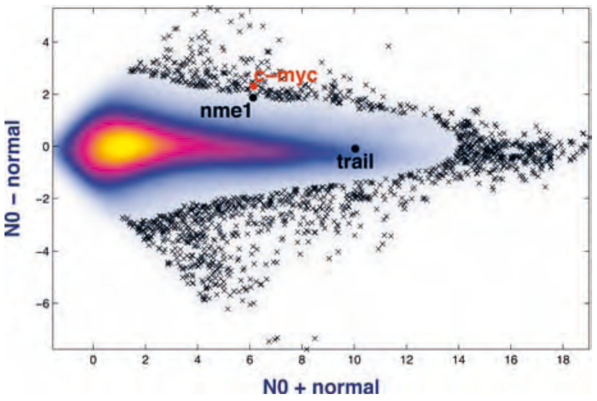


Figure 7 Scatterplot of a pairwise comparison of noncancerous colon tissue and a colorectal tumor. Individual probes are represented by crosses. The x-coordinate is the average of the appropriately calibrated and transformed intensities (see Section 5.2). The y-coordinate is their difference and is a measure of differential transcription. The array used in this experiment contained 152 000 probes representing around 70 000 different clones. Since plotting all of these would lead to an uninformative solid black blob in the center of the plot, the point density is visualized by a color scale and only 1500 data points in sparser regions are individually plotted.

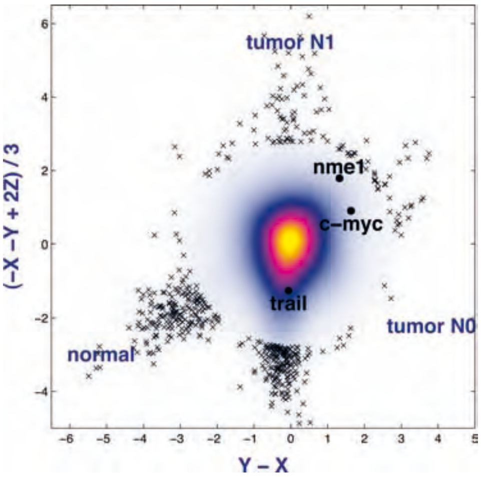


Figure 8 Scatterplot of a triple comparison between noncancerous colon tissue, a lymph node-negative colorectal tumor (N0) and a lymph node-positive tumor (N1). The measurements from each probe correspond to a point in three-dimensional space and are projected orthogonally on a plane perpendicular to the (1,1,1)-axis. The three coordinate axes of the data space correspond to the vectors from the origin of the plot to the three labels "normal", "tumor N0" and "tumor N1". The (1,1,1)-axis corresponds to average intensity, while differences between the three tissues are represented by the position of the measurements in the two-dimensional plot plane. For instance, both c-*myc* and *nme1* (nucleoside diphosphate kinase A) are higher transcribed in the N0 and in the N1 tumor, compared to the noncancerous tissue. However, while the increase is approximately balanced for c-*myc* in the two tumors, *nme1* is more upregulated in the N1 tumor than in the N0 tumor, a behavior that is consistent with a gene involved in tumor progression. On the other side, the apoptosis-inducing receptor *trail-r2* is downregulated specifically in the N1 tumors, while it has about the same intermediate-high transcription level in the noncancerous tissue and the N0 tumor. Similar behavior of these genes was observed over repeated experiments.

microarray measurement corresponds to average intensity and hence is not directly informative with respect to differential transcription. Note that if the plotted data was preprocessed through a variance-stabilizing transformation, its variance does not depend on the $(1, 1, 1)$-component.

2.4 Batch Effects

Present day microarray technology measures abundances only in terms of relative probe intensities and generally provides no calibration to absolute physical units. Hence, the comparison of measurements between different studies is difficult. Moreover, even within a single study, the measurements are highly susceptible to *batch effects*. By this term, we refer to experimental factors that (i) add systematic biases to the measurements and (ii) may vary between different subsets or stages of an experiment. Some examples are [33]:

(i) *Spotting.* To manufacture spotted microarrays, the probe DNA is deposited on the surface through spotting pins. Usually, the robot works with multiple pins in parallel, and the efficiency of their probe delivery may be quite different [e.g. Figure 4(d) or [11]]. Furthermore, the efficiency of a pin may change over time through mechanical wear and the quality of the spotting process as a whole may be different at different times, due to varying temperature and humidity conditions.

(ii) *PCR amplification.* For cDNA arrays, the probes are synthesized through PCR, whose yield varies from instance to instance. Typically, the reactions are carried out in parallel in 384-well plates, and probes that have been synthesized in the same plate tend to have correlated variations in concentration and quality. An example is shown in Figure 9.

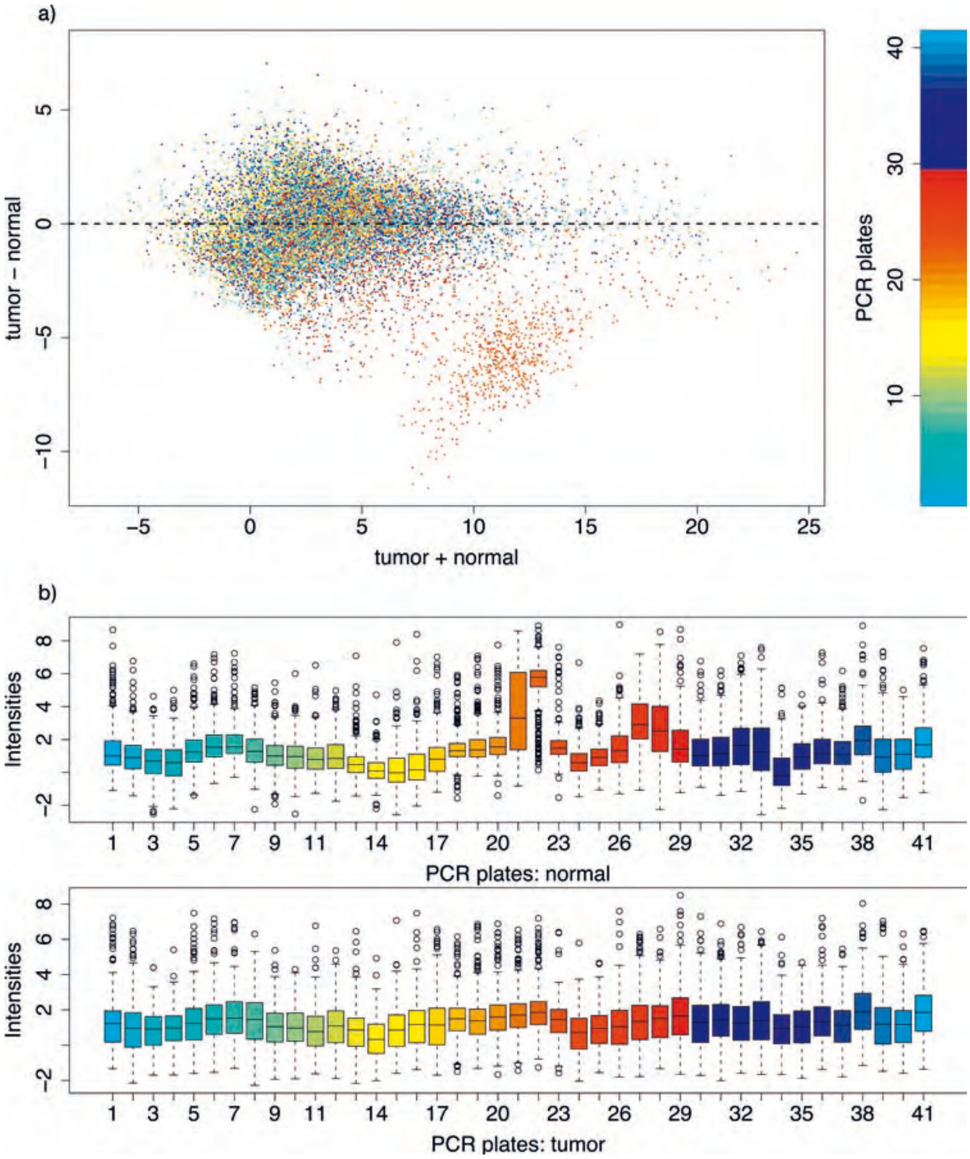
(iii) *Sample preparation protocols.* Reverse transcription and labeling are complex biochemical reactions, whose efficiencies are variable and may depend sensitively on a number of hard-to-control circumstances. Furthermore, RNA can quickly degrade, hence the outcome of the experiment can depend sensitively on when and how conditions that prevent RNA degradation are applied to the tissue samples.

(iv) *Array coating.* Both the efficiency of the probe fixation on the array, as well as the amount of unspecific background fluorescence strongly depend on the array coating.

(v) *Scanner and image analysis.* Different scanners can produce slightly different intensity images even from identical slides and the performance of the same scanner can drift over time. Different image analysis programs can use different algorithms to calculate probe summaries and the same program, in particular when it requires human interaction, can produce different results from the same image.

These considerations have important consequences for the experimental design. First, any variation that can be avoided by any means within an experiment should be avoided. Second, any variation that cannot be avoided should be organized in such a manner that it does not confound the biological question of interest. Clearly, when looking for differences between two tumor types, it would not be wise to have samples of one tumor type processed by one laboratory, and samples of the other type by another laboratory.

Points (i) and (ii) are specific for spotted cDNA arrays. To be less sensitive against these variations, the two-color-labeling protocol is used, which employs the simultaneous hybridization of two samples to the same array [12]. Ideally, if only ratios of intensities between the two color channels are considered, variations in probe abundance should cancel out. Empirically,


a)
tumor – normal
5
0
−5
−10
−5
0
5
10
15
20
25
tumor + normal
PCR plates
10
20
30
40
b)
Intensities
8
6
4
2
−2
1 3 5 7 9 11 14 17 20 23 26 29 32 35 38 41
PCR plates: normal
Intensities
8
6
4
2
−2
1 3 5 7 9 11 14 17 20 23 26 29 32 35 38 41
PCR plates: tumor

they do not quite do so, which may, for example, be attributed to the fact that observed intensities are the sum of probe-specific signal and unspecific background [43]. Furthermore, in the extreme case of total failure of the PCR amplification or the DNA deposition for probes on some, but not all arrays in an experimental series, artifactual results are hardly avoidable.

If any of the factors (iii)–(v) is changed within an experiment, there is a good chance that this will show up later in the data as one of the most pronounced sources of variation. A simple and instructive visual tool for exploring such variations is the correlation plot: given a set of d arrays, each represented through a high-dimensional vector $\vec{Y}_i$ of suitably transformed and filtered probe intensities, calculate the $d \times d$ correlation matrix $\mathrm{corr}(\vec{Y}_i, \vec{Y}_j)$, sort its rows and columns according to different experimental factors, and visualize the resulting false-color images.

2.5 Along Chromosome Plots

Visualization of microarray data along genomic coordinates can be useful for many purposes, e.g. to detect genomic aberrations (deletions, insertions) or regulatory mechanisms that act at the level of genomic regions [35]. Here, we show an example from the application of a genome tiling microarray to transcription analysis.

While conventional microarrays contain only a preselected set of one or a few probes for each of a set of known or putative transcripts, more recent microarray designs provide probes for the complete genomic sequence content of an organism. Rather than relying on a manufacturer's assignment of probes to genes (or, more exactly, target transcripts), it can become part of the analysis to make the assignment on the basis of the data themselves. An example is shown in Figure 10.

←

Figure 9 (a) Scatterplot of logarithmized intensities from a pair of single-color cDNA arrays, comparing renal cell carcinoma to matched noncancerous kidney tissue. Similar to Figure 7, the x-coordinate represents average and the y-coordinate differential signal. In the bottom of the plot, there is a cloud of probes that appear to represent a cluster of strongly downregulated genes. However, closer scrutiny reveals that this is an experimental artifact: (b) shows the box plots of the intensities for the two arrays, separately for each of the 41 PCR plates (see text). Probes from plates no. 21, 22, 27 and 28 have extraordinarily high intensities on one of the arrays, but not on the other. Since the clone selection was quasi-random, this points to a defect in the probe synthesis that affected one array, but not the other. The discovery of such artifacts may be facilitated by coloring the dots in the scatterplot by attributes such as PCR plate of origin or spotting pin. While the example presented here is an extreme one, caution towards batch artifacts is warranted whenever arrays from different manufacturing lots are used in a single study.

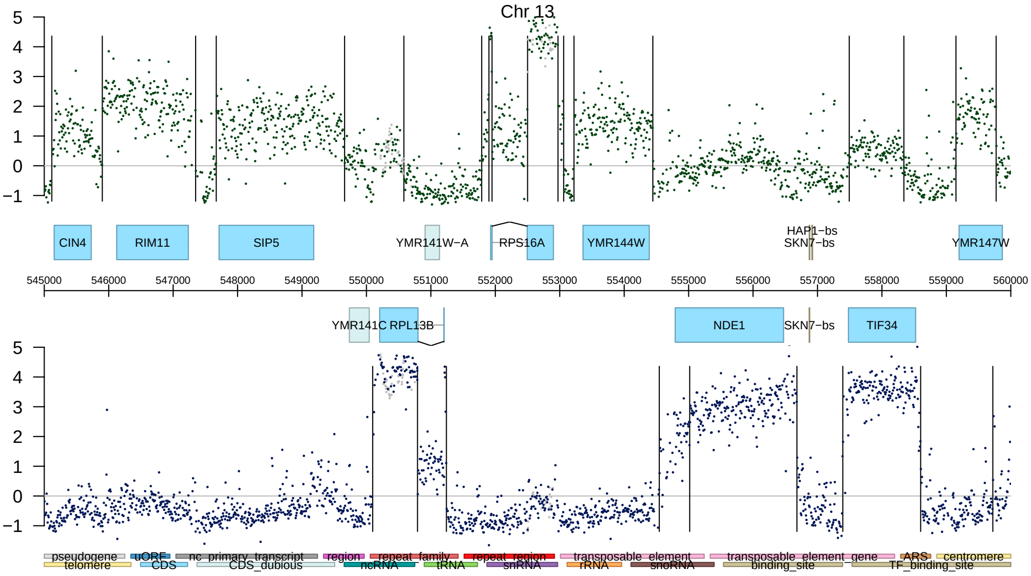


Figure 10 Along chromosome plot of the data from an Affymetrix genechip that contains 25-mer oligonucleotide probes covering the whole genome of *Saccharomyces cerevisiae* in steps of 8 bases, on both strands. The displayed values are the base 2 logarithms of the ratios between intensities from hybridization with a poly(A) RNA sample and with a genomic DNA sample. Also shown are genomic coordinates and annotated genomic features. The vertical bars show the segmentation of the intensity signal into an approximately piecewise constant function [9a, 18a, 28]. The data allows for the mapping of $5'$ and $3'$ untranslated regions, the deconvolution of populations of overlapping transcripts of different lengths, and the detection of novel transcripts.

2.6 Sensitivity and Specificity of Probes

The probes on a microarray are intended to measure the abundance of the particular transcript or locus that they are assigned to. However, probes may differ in terms of their sensitivity and specificity. Here, sensitivity means that a probe's fluorescence signal indeed responds to changes in the abundance of its target; specificity means that it does not respond to other targets or other types of perturbations.

Probes may lack sensitivity. Some probes initially identified with a gene do not actually hybridize to any of its products. Some probes will have been developed from information that has been superseded. In some cases, the probe may correspond to a different gene or it may in fact not represent any gene. In other cases, a probe may match only certain transcript variants of a given gene, which makes it more complicated to derive statements on the gene's expression (e.g. NDE1 and CIN4 in Figure 10). There is also the possibility of human error [15, 24].

A potential problem, especially with short oligonucleotide technology, is that the probes may not be specific, i.e. in addition to matching the intended transcript, they may also match others. In this case, we expect the observed intensity to be a composite from all matching transcripts. Note that, particularly in the case of higher eukaryotes, we are limited by the current state of knowledge of the transcriptomes. As our knowledge improves, the information about specificity of probes should also improve.

3 Error Models

3.1 Motivation

With a microarray experiment, we aim to make statements about the abundances of specific molecules in a set of biological samples. However, the quantities that we measure are the fluoresence intensities of the different elements of the array. The measurement process consists of a cascade of biochemical reactions and an optical detection system with a laser scanner or a CCD camera. Biochemical reactions and detection are performed in parallel, allowing millions of measurements on one array. Subtle variations between arrays, the reagents used and the environmental conditions lead to slightly different measurements even for the same sample.

The effects of these variations may be grouped in two classes. *Systematic effects* affect a large number of measurements (e.g. the measurements for all probes on one array or the measurements from one probe across several arrays) simultaneously. Such effects can be estimated and, to good approximation, be removed. Other kinds of effects are random, with no well-understood pattern. These effects are commonly called *stochastic effects* or *noise*. This classification is not a property of the variations *per se*, but rather reflects our understanding of them and our modeling effort. The same kind of variation can be considered stochastic in one analysis and systematic in another.

So what is the purpose of constructing error models for microarrays? There are three aspects as outlined below.

3.1.1 **Obtaining Optimal Estimates**

Stochastic models are useful for preprocessing because they permit us to find *optimal* estimates of the systematic effects. We are interested in estimates that are precise and accurate. However, given the noise structure of the data we sometimes have to sacrifice accuracy for better precision and *vice versa*. An appropriate stochastic model will aid in understanding the accuracy-precision, or bias-variance, trade-off.

3.1.2 **Biological Inference**

Stochastic models are also useful for statistical inference from experimental data. Consider an experiment in which we want to compare gene expression in the colons of mice that were treated with a substance and mice that were not. If we have many measurements, we can simply compare their empirical distributions. For example, if the values from 10 replicate measurements for the DMBT1 gene in the treated condition are all larger than 10 measurements from the untreated condition, the Wilcoxon test tells us that with a p-value of 10^{-5} the level of the transcript is really elevated in the treated mice. However, often it is not possible, too expensive or unethical to obtain so many replicate measurements for all genes and for all conditions of interest. Often, it is also not necessary. If we have some confidence in a model, we are able to draw significant conclusions from fewer replicates.

3.1.3 **Quality Control**

Quality control is yet another example of the usefulness of stochastic models: if the distribution of a new set of data greatly deviates from the model, this may direct our attention to quality issues with these data.

3.2 The Additive–Multiplicative Error Model

3.2.1 **Induction from Data**

Different hybridizations will result in more or less different signal intensities even if the biological sample is the same. To see this, let us look in Figure 11 at the empirical distribution of the intensities from six replicate Affymetrix genechips. The data are part of the Latin Square Data for Expression Algorithm Assessment provided by Affymetrix (http://www.affymetrix.com/support/technical/sample_data/datasets.affx).

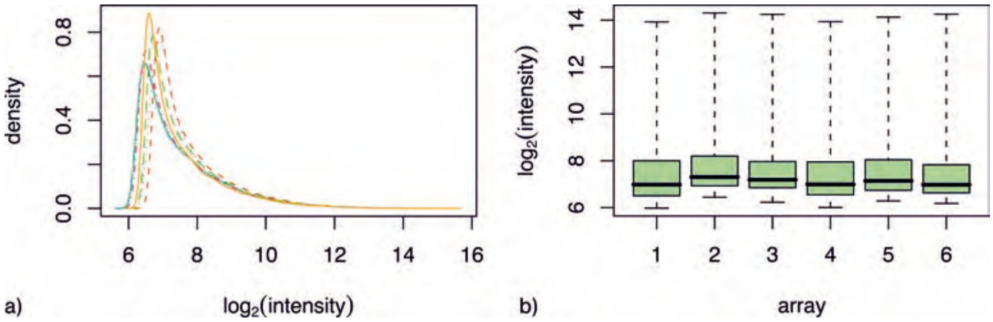


Figure 11 (a) Density estimates of probe intensity data from six replicate Affymetrix arrays. The x-axis is on a logarithmic scale (base 2). (b) Box plots of the same data.

One task of error modeling is to deal with background noise. Notice in Figure 11 that the smallest values attained are around 64, with slight differences between the arrays. We know that many of the probes are not supposed to be hybridizing to anything (as not all genes are expressed), so many measurements should indeed be 0. A bottom-line effect of not removing background noise is that estimates of differential expression are biased. Specifically, the ratios are attenuated toward 1. This can be seen using the Affymetrix spike-in experiment, where genes were spiked in at known concentrations. Figure 12(a) shows the observed concentrations versus nominal concentrations of the spiked-in genes. Measurements with smaller nominal concentrations appear to be affected by attenuation bias. To see why, notice that the curve has a slope of about 1 for high nominal concentrations, but becomes flat as the nominal concentration gets closer to 0. This is consistent with the additive background noise model which we will discuss in the next section. Mathematically, it is easy to see that if s_1/s_2 is the true ratio, and b_1 and b_2 are approximately equal positive numbers, then $(s_1 + b_1)/(s_2 + b_2)$ is closer to 1 than the true ratio, and the more so the smaller the s_i are compared to the b_i.

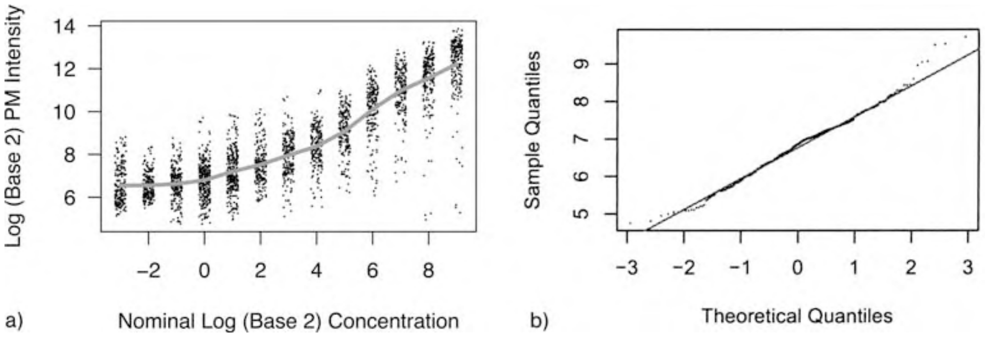


Figure 12 (a) Plot of observed against nominal concentrations. Both axes are on the logarithmic scale (base 2). The curve represents the average value of all probes at each nominal concentration. Nominal concentrations are measured in picomoles. (b) Normal quantile–quantile plot of the logarithmic (base 2) intensities for all probes with the same nominal concentration of 1 pmol.

Figure 12(b) shows a normal quantile–quantile plot of logarithmic intensities of probes for genes with the same nominal concentration. Note that these appear to roughly follow a normal distribution. Figure 12 supports the multiplicative error assumption of the model that we formulate in the next section.

3.2.2 A Theoretical Deduction

Consider the generic observation equation $z = f(x, y)$, where z is the outcome of the measurement, x is the true underlying quantity that we want to measure, the function f represents the measurement apparatus and $y = (y_1, \ldots, y_n)$ is a vector that contains all other parameters on which the functioning of the apparatus may depend. The functional dependence of f on some of the y_i may be known; on others, it may not. Some of the y_i are explicitly controlled by the experimenter; some are not. For a well-constructed measurement apparatus, f is a well-behaved, smooth function and we can rewrite the observation equation as:

$$z = f(0, y) + f'(0, y)\, x + O(x^2), \tag{1}$$

where $f(0, y)$ is the baseline value that is measured if x is zero, f' is the derivative of f with respect to x, $f'(0, y)$ is a gain factor and $O(x^2)$ represents nonlinear efffects. By proper design of the experiment, the nonlinear terms can be made negligibly small within the relevant range of x. Examples for the parameters y in the case of microarrays are the efficiencies of mRNA extraction, reverse transcription, labeling and hybridization reactions, amount and quality of probe DNA on the array, unspecific hybridization, dye quantum yield, scanner gain, and background fluorescence of the array.

Ideally, the parameters y could be fixed once and forever exactly at some value $\bar{y} = (\bar{y}_1, \ldots, \bar{y}_n)$. In practice, they will fluctuate around $\bar{y}$ between repeated experiments. If the fluctuations are not too large, we can expand:

$$f(0, y) \approx f(0, \bar{y}) + \sum_{i=1}^{n} \frac{\partial f(0, \bar{y})}{\partial y_i} (y_i - \bar{y}_i) \tag{2}$$

$$f'(0, y) \approx f'(0, \bar{y}) + \sum_{i=1}^{n} \frac{\partial f'(0, \bar{y})}{\partial y_i} (y_i - \bar{y}_i). \tag{3}$$

The sums on the right-hand sides of Eqs. (2) and (3) are linear combinations of a large number n of random variables with mean 0. Thus, it is a reasonable approximation to model $f(0, y)$ and $f'(0, y)$ as normally distributed random variables with means $a = f(0, \bar{y})$ and $b = f'(0, \bar{y})$ and variances σ_a^2 and σ_b^2, respectively. Thus, omitting the nonlinear term, Eq. (1) leads to:

$$z = a + \varepsilon + b\, x(1 + \eta), \tag{4}$$

with $\varepsilon \sim N(0, \sigma_a^2)$ and $\eta \sim N(0, \sigma_b^2 / b^2)$. This is the *additive–multiplicative error model* for microarray data, which was proposed by Ideker and coworkers [29]. Rocke and Durbin [29] proposed it in the form:

$$z = a + \varepsilon + b\, x \exp(\eta), \tag{5}$$

which is equivalent to Eq. (4) up to first-order terms in η. Models (4) and (5) differ significantly only if the coefficient of variation σ_b/b is large. For microarray data, it is typically smaller than 0.2, thus the difference is of little practical relevance.

One of the main predictions of the error model (4) is the form of the dependence of the variance of z on its mean $\mathrm{E}(z)$:

$$\mathrm{Var}(z) = v_0^2 + \frac{\sigma_b^2}{b^2}\,(\mathrm{E}(z) - z_0)^2\,, \tag{6}$$

that is, a strictly positive quadratic function. In the following we will assume that the correlation between ε and η is negligible. Then the parameters of Eq. (6) are related to those of Eq. (4) via $v_0^2 = \sigma_a^2$ and $z_0 = a$. If the correlation is not negligible, the relationship is slightly more complicated, but the form of Eq. (6) remains the same.

4 Normalization

A parametrization of Eq. (5) that captures the main factors that play a role in current experiments is:

$$z_{ip} = a_{i,s(p)} + \varepsilon_{ip} + b_{i,s(p)} B_p\, x_{j(i),k(p)} \exp(\eta_{ip}). \tag{7}$$

Let us dissect this equation. The index p labels the different probes on the array and $k = k(p)$ is the transcript or locus that probe p maps to. Each probe is intended to map to exactly one k, but one transcript or locus may be represented by several probes. B_p is the probe-specific gain factor of the p-th probe. i counts over the arrays and, if applicable, over the different dyes. $j = j(i)$ labels the biological conditions (e.g. normal/diseased). $a_{i,s(p)}$ and $b_{i,s(p)}$ are normalization offsets and scale factors that may be different for each i and possibly for different groups ("strata") of probes $s = s(p)$. Probes can be stratified according to their physicochemical properties [39] or array-manufacturing parameters such as print-tip [41] or spatial location. In the simplest case, $a_{i,s(p)} = a_i$ and $b_{i,s(p)} = b_i$ are the same for all probes on an array. The noise terms ε and η are as above.

On an abstract level, much of the literature on normalization can be viewed as an application of Eq. (7) to data, employing various choices for probe stratification, making simplifying assumptions on some of its parameters, rearranging the equation and using different, more or less robust algorithms to estimate its parameters [2, 16–18, 21, 22, 25, 33, 38].

There is an alternative approach to normalization, which focuses on non-parametric methods and the algorithmic aspects. In this approach, one identifies those statistics (properties) of the data that one would like to be the same, say, between different arrays, but observes empirically in the raw data that they are not. One then designs an algorithm that transforms the data so that the desired statistics are made the same in the normalized data. The intention is that the interesting, biological signal is kept intact in the process [4,31,41].

For example, the *loess* normalization [41] calculates log-ratios M between the red and the green intensities on one array, plots them versus A, the logarithm of the geometric mean (see Figures 6 and 7), and postulates that a nonparametric regression line, calculated by a so-called loess scatterplot smoother [8] ought to look straight. In order to achieve this, the loess-fitted regression values for M are subtracted from the observed values and the residuals are kept as the normalized data.

In *quantile* normalization, one plots the histogram of log-transformed intensities for each array and postulates that they all should look the same. Bolstad and coworkers [4] have introduced an algorithm that achieves this by rank-transforming the data and then mapping the ranks back to a consensus distribution. The result is a monotonous nonlinear transformation for each array which assures that the distribution function of the transformed data is the same for all arrays.

These nonparametric methods are popular because they always "work", by construction, and usually in a fully automatic manner. In contrast, in model-based approaches it may turn out that a given set of data does not fit. Also, the assessment of goodness of fit is not easily automated, and often requires some human interaction. If the fit is bad, the data cannot be normalized, thus not be further analysed, and an expensive and time-consuming experiment would be left hanging.

However, there is a caveat: experiments may contain failed hybridizations, degraded samples and nonfunctioning probes. The goodness-of-fit criteria from a model-based normalization method can serve as relevant criteria to detect these. With a method that "always works", there is the risk of overlooking these aspects of the data, to normalize them away and move on to further analysis pretending that everything was fine. Conversely, one needs a sophisticated and largely nonautomatic quality control step. So, if we consider normalization and quality control together as one task, the balance between model-based and nonparametric methods is more even.

Furthermore, parametric methods have, if they are appropriate, better power than nonparametric ones. They provide better sensitivity and specificity in the application of detecting differentially expressed genes. Given the typically small sample size and the expense of microarray experiments, this is a consequential point. It has been verified in comparison studies [9,16].

5 Detection of Differentially Expressed Genes

5.1 Stepwise versus Integrated Approaches

Most commonly used is the *stepwise* approach to microarray data analysis. It takes a collection of raw data as input and produces an *expression matrix* as output. In this matrix, rows correspond to gene transcripts and columns to conditions. Each matrix element represents the abundance, in certain units, of a transcript under a condition. Subsequent biological analyses work off the expression matrix and generally do not consider the raw data. The preprocessing itself is largely independent of the subsequent biological analysis. In some cases, the preprocessing is further subdivided into a set of sequential instructions, e.g. subtract the background, then normalize the intensities, then summarize replicate probes, then summarize replicate arrays. By its *modularity*, the stepwise approach allows us to structure the analysis workflow. Software, data structures and methodology can be more easily reused. For example, the same machine learning algorithm can be applied to an expression matrix irrespective of whether the raw data were obtained on Affymetrix chips or on spotted cDNA arrays. A potential disadvantage of the stepwise approach is that each step is optimized for itself and that the results of subsequent steps have no influence on the previous ones. For example, the normalization procedure has to deal with whatever the preceding background correction procedure produced and has no chance to ask it to reconsider. This can and does lead to inefficiencies.

In contrast, *integrated* approaches try to gain sensitivity by doing as much as possible at once, and therefore using the available data more efficiently. For example, rather than calculating an expression matrix, one might fit an ANOVA-type linear model that includes both technical covariates, such as dye and sample effects, and biological covariates, such as treatment [22], to the raw data. In Ben Bolstad's affyPLM method [5], the weighting and summarization of the multiple probes per transcript on Affymetrix chips is integrated with the detection of differential expression. Another example is the vsn method [16], which integrates background subtraction and normalization.

Stepwise approaches are often presented as modular data-processing pipelines; integrated approaches as statistical models whose parameters are to be fitted to the data. In practice, data analysts will often choose to use a combination of both approaches, maybe starting with the stepwise approach, do a first round of high-level analyses and then turn back to the raw data to answer specific questions that arise. Good software tools allow us to use and explore both stepwise and integrated methods, and to freely adapt and combine them.

5.2 Measures of Differential Eexpression: The Variance Bias Trade-off

What is a good statistic to compare two (or several) measurements from the same probe on a microarray, taken from hybridizations with different biological targets?

Plausible choices include the difference, the ratio and the logarithm of the ratio. To understand the problem more systematically, we return to the notation of Section 3.2.2. Let z_1 and z_2 denote two measurements from the same probe, and assume that they are distributed according to Eq. (5) with the same parameters a, b, σ_a, and σ_b, but possibly with different values of x_1, x_2, corresponding to different levels of the target in the biological samples of interest. We want to find a function $h(z_1, z_2)$ that fulfills the following two conditions: *antisymmetry*, $h(z_1, z_2) = -h(z_2, z_1)$ for all x_1, x_2 and *homoskedasticity*, constant variance of $h(z_1, z_2)$ independent of x_1, x_2. An approximate solution is given by [17]:

$$h(z_1, z_2) = \operatorname{arsinh}\left(\frac{z_1 - a}{\beta}\right) - \operatorname{arsinh}\left(\frac{z_2 - a}{\beta}\right), \qquad (8)$$

with $\beta = \sigma_a b / \sigma_b$. If both z_1 and z_2 are large, this expression approaches the log-ratio:

$$q(z_1, z_2) = \log(z_1 - a) - \log(z_2 - a). \qquad (9)$$

However, for $z_i \to a$, the log-ratio $q(z_1, z_2)$ has a large, diverging variance, a singularity at $z_i = a$ and is not defined in the range of real numbers for $z_i < a$.

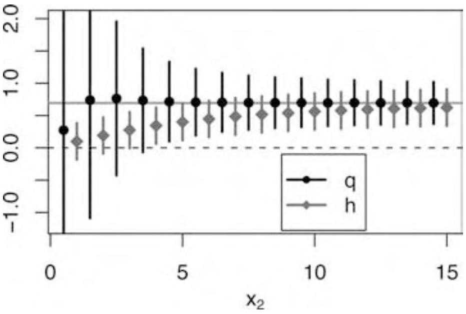


Figure 13 The shrinkage property of the generalized log-ratio h. Blue diamonds and error bars correspond to mean and standard deviation of $h(z_1, z_2)$, *cf.* Eq. (8); black dots and error bars to $q(z_1, z_2)$, cf. Eq. (9). Data were generated according to Eq. (5) with $x_2 = 0.5, \ldots, 15$, $x_1 = 2x_2$, $a = 0$, $\sigma_a = 1$, $b = 1$, $\sigma_b = 0.1$. The horizontal line corresponds to the true log-ratio $\log(2) \sim 0.693$. For intensities x_2 that are larger than about 10 times the additive noise level σ_a, h and q are approximately equal. For smaller intensities, we can see a *variance bias trade-off*: q has no bias but a huge variance, thus an estimate of the fold change based on a limited set of data can be arbitrarily off. In contrast, h keeps a constant variance – for the price of systematically underestimating the true fold change.

These unpleasant properties are important for applications: many genes are not expressed or only weakly expressed in some, but not all conditions of interest. That means we need to compare conditions in which, for example, x_1 is large and x_2 is small. The log-ratio (9) is not a useful quantity for this purpose, since the second term will wildly fluctuate and be sensitive to small errors in the estimation of the parameter a. In contrast, the statistic (8), which is called the *generalized log-ratio* [30], is well-defined everywhere and robust against small errors in a. It is always smaller in magnitude than the log-ratio (see also Figure 13):

$$\begin{aligned} &|h(z_1, z_2)| < |q(z_1, z_2)| \qquad \forall z_1, z_2, \\ &h(z_1, z_2) \approx q(z_1, z_2) \qquad \text{for } z_1, z_2 \gg a + \beta. \end{aligned} \tag{10}$$

The exponentiated value:

$$\widehat{FC} = \exp(h(z_1, z_2)), \tag{11}$$

can be interpreted as a shrinkage estimator for the *fold-change* x_1/x_2. It is more specific, i.e. leads to fewer false positives in the detection of differentially expressed genes, than the naive estimator $(z_1 - a)/(z_2 - a)$ [13,16].

5.3 Identifying Differentially Expressed Genes from Replicated Measurements

One of the main motivations for performing microarray studies is the need to identify genes whose patterns of expression differ according to phenotype or experimental condition. Gene expression is a well-coordinated system and, hence, measurements on different genes are in general not independent. Given more complete knowledge of the specific interactions and transcriptional controls it is conceivable that meaningful comparisons between samples can be made by considering the joint distribution of specific sets of genes. However, the high dimension of gene expression space prohibits a comprehensive exploration, while the fact that our understanding of biological systems is only in its infancy means that in many cases we do not know which relationships are important and should be studied. In current practice, differential expression analysis will therefore at least start with a gene-by-gene approach, ignoring the dependencies between genes.

A simple approach in the comparison of different conditions is to rank genes by the difference of means of appropriately transformed intensities in the sense of Section 5.2. This may be the only possibility in cases where no, or very few replicates, are available. An analysis solely based on a difference of means statistic, however, does not allow the assessment of significance of expression differences in the presence of biological and experimental variation, which may differ from gene to gene. This is the main reason for using statistical tests to assess differential expression. Generally, one might look at various properties of the distributions of a gene's expression levels under different conditions, though most often location parameters of these distributions, such as the mean or the median, are considered. One may distinguish between parametric tests, such as the t-test, and nonparametric tests, such as the Mann–Whitney test or permutation tests. Parametric tests usually have a higher power if the underlying model assumptions, such as normality in the case of the t-test, are at least approximately fulfilled. Nonparametric tests do have the advantage of making less stringent assumptions on the data-generating distribution. In many microarray studies however, a small sample size leads to insufficient power for nonparametric tests and, as discussed in Section 3.1, increasing the sample size might be uneconomical or unethical if parametric alternatives are feasible. A pragmatic approach in these situations is to employ parametric tests, but to use the resulting p-values cautiously to rank genes by their evidence for differential expression, rather than taking them for the truth.

A generalized log-transformation of intensity data as described in Section 5.2 can be beneficial not only when using a difference of means statistic, but also for parametric statistical tests. Typically it will make the distribution of replicated measurements per gene roughly symmetric and more or less close to normal. The variance stabilization achieved by the transformation can be advantageous for gene-wise statistical tests that rely on variance homogeneity, because it diminishes differences in variance between experimental conditions that are due to differences in the intensity level; however, of course differences in variance between conditions may also have gene-specific biological reasons, and these will remain untouched by the transformation.

One or two group t-test comparisons, multiple group ANOVAs and more general trend tests are all instances of linear models that are frequently used for assessing differential gene expression. As a parametric method, linear modeling is subject to the caveats discussed above, but the convenient interpretability of the model parameters often makes it the method of choice for microarray analysis. Due to the aforementioned lack of information regarding coregulation of genes, linear models are generally computed for each gene separately. When the genes of interest are identified, investigators can hopefully begin to study their coordinated regulation for more sophisticated modeling of their joint behavior.

The approach of conducting a statistical test for each gene is popular, largely because it is relatively straightforward and a standard repertoire of methods can be applied. However, the approach has a number of drawbacks – most important is the fact that a large number of hypothesis tests is carried out, potentially leading to a large number of falsely significant results. Multiple testing procedures allow us to assess the overall significance of the results of a family of hypothesis tests. They focus on specificity by controlling type I (false-positive) error rates such as the *familywise error rate* or the *false discovery rate* [10]. Still, multiple hypothesis testing remains a problem, because an increase in specificity, as provided by p-value adjustment methods, is coupled with a loss of sensitivity, i.e. a reduced chance of detecting true positives. Furthermore, the genes with the most drastic changes in expression are not necessarily the "key players" in the relevant biological processes [37]. This problem can only be addressed by incorporating prior biological knowledge into the analysis of microarray data, which may lead to focusing the analysis on a specific set of genes. Also if such a biologically motivated preselection is not feasible, the number of hypotheses to be tested can often be reasonably reduced by nonspecific filtering procedures, discarding, e.g. genes with consistently low intensity values or low variance across the samples. This is especially relevant in the case of genome-wide arrays, as often only a minority of all genes will be expressed at all in the cell type under consideration.

Many microarray experiments involve only few replicates per condition, which makes it difficult to estimate the gene-specific variances that are used, e.g. in the t-test. Different methods have been developed to exploit the variance information provided by the data of all genes [1, 20, 27, 36]. In Ref. [34], an empirical Bayes approach is implemented that employs a global variance estimator s_0^2 computed on the basis of all genes' variances. The resulting test statistic is a moderated t-statistic, where instead of the single-gene estimated variances s_g^2, a weighted average of s_g^2 and s_0^2 is used. Under certain distributional assumptions, this test statistic can be shown to follow a t-distribution under the null hypothesis with the degrees of freedom depending on the data.

6 Software

Many of the algorithms and visualizations discussed in this chapter are available through the Bioconductor project [14]. This project is an initiative for the collaborative creation of extensible software for computational biology and bioinformatics. Its goals include fostering development and widespread use of innovative software, reducing barriers to entry into interdisciplinary scientific research, and promoting the achievement of remote reproducibility of research results.

The software produced by the Bioconductor project is organized into packages, each of which is written and maintained relatively autonomously by its authors, who come from many different institutions around the world, and which are held together through a common language platform, R, a set of common data structures, a uniform structure of package organization and documentation, and a lively user community.

Results of this project are available on the website http://www.bioconductor.org. Among the packages that are most relevant for the subject of this chapter are affy (preprocessing of Affymetrix genechip data), vsn (affine-linear parametric normalization and variance stabilizing normalization), marray (two-color preprocessing) and limma (differential expression with linear models). Some further aspects are represented by arrayMagic (high-throughput quality control and preprocessing) and tilingArray (along chromosome plots and segmentation).

Acknowledgments

We thank Günther Sawitzki, Robert Gentleman, Dirk Buschmann and Andreas Buneß for fruitful discussions. We are grateful to all contributors to the R and Bioconductor projects for sharing their software. Holger Sültmann, Annemarie Poustka, Anke Schroth, Lars Steinmetz and Lior David are gratefully acknowledged for sharing experimental data.

References

1 BALDI, P. AND A. D. LONG. 2001. A Bayesian framework for the analysis of microarray expression data: regularized t-test and statistical inferences of gene changes. *Bioinformatics* **17**:509–19.

2 BEISSBARTH, T., K. FELLENBERG, B. BRORS, et al. 2000. Processing and quality control of DNA array hybridization data. *Bioinformatics* **16**:1014–22.

3 BERTONE, P., V. STOLC, T. E. ROYCE, et al. 2004. Global identification of human transcribed sequences with genome tiling arrays. *Science* **306**:2242–46.

4 BOLSTAD, B. M., R. A. IRIZARRY, M. ASTRAND, et al. 2003. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* **19**:185–93.

5 BOLSTAD, B. M., F. COLLIN, K. M. SIMPSON, R. A. IRIZARRY AND T. P. SPEED. 2004. Experimental design and low-level analysis of microarray data. Int. Rev. Neurobiol. **60**: 25–58.

6 CHENG, J., P. KAPRANOV, J. DRENKOW, et al. 2005. Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. *Science* **308**:1149–54.

7 CHURCHILL, G. 2002. Fundamentals of experimental design for cDNA microarrays. *Nature Genet.* **32 (Suppl. 2)**:490–95.

8 CLEVELAND, W., E. GROSSE AND W. SHYU. 1992. Local regression models. In CHAMBERS, J. M. AND HASTIE, T. J. (eds.), *Statistical Models*. Wadsworth & Brooks, Pacific Grove: 309–76.

9 COPE, L. M., R. A. IRIZARRY, H. A. JAFFEE, Z. WU AND T. P. SPEED. 2004. A benchmark for Affymetrix GeneChip expression measures. *Bioinformatics* **20**:323–31.

9a DAVID, L., W. HUBER, M. GRANOVSKAIA ET AL. 2006. A high-resolution map of transcription in the yeast genome. Proc. Natl. Acad. Sci. USA **103(14)**: 5320–5.

10 DUDOIT, S., J. P. SHAFFER AND J. C. BOLDRICK. 2003. Multiple hypothesis testing in microarray experiments. *Stat. Sci.* **18**:71-103.

11 DUDOIT, S., Y. H. YANG, T. P. SPEED AND M. J. CALLOW. 2002. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Stat. Sinica* **12**:111–39.

12 DUGGAN, D. J., M. BITTNER, Y. CHEN, P. MELTZER AND J. M. TRENT. 1999. Expression profiling using cDNA microarrays. *Nat. Genet.* **21 (Suppl. 1)**:10–4.

13 DURBIN, B. P., J. S. HARDIN, D. M. HAWKINS AND D. M. ROCKE. 2002. A Variance-Stabilizing Transformation for Gene-expression Microarray Data. *Bioinformatics* **18 (Suppl. 1)**:S105–10.

14 GENTLEMAN, R. C., V. J. CAREY, D. J. BATES, et al. 2004. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biol.* **5**:R80.

15 HALGREN, R. G., M. R. FIELDEN, C. J. FONG AND T. R. ZACHAREWSKI. 2001. Assessment of clone identity and sequence fidelity for 1189 IMAGE cDNA clones. *Nucleic Acids Res.* **29**:582–88.

16 HUBER, W., A. VON HEYDEBRECK, H. SÜLTMANN, A. POUSTKA AND M. VINGRON. 2002. Variance Stabilization Applied to Microarray Data Calibration and to the Quantification of Differential Expression. *Bioinformatics* **18 (Suppl. 1)**:S96–104.

17 HUBER, W., A. VON HEYDEBRECK, H. SÜLTMANN, A. POUSTKA AND M. VINGRON. 2003. Parameter estimation for the calibration and variance stabilization of microarray data. *Stat. Appl. Genet. Mol. Biol.* **2**:article 3.

18 HUBER, W. 2005. *Robust Calibration and Variance Stabilization with vsn.* http://www.bioconductor.org

18a HUBER, W., J. TOEDLING AND L. M. STEINMETZ. 2006. Transcript mapping with high-density oligonucleotide tiling arrays. Bioinformatics **22(16)**: 1963–70.

19 IDEKER, T., V. THORSSON, A. SIEGEL AND L. HOOD. 2000. Testing for differentially expressed genes by maximum-likelihood analysis of microarray data. *J. Comput. Biol.* **7**:805–18.

20 KENDZIORSKI, C., M. NEWTON, H. LAN AND M. GOULD. 2003. On parametric empirical Bayes methods for comparing multiple groups using replicated gene expression profiles. *Stat. Med.* **22**:3899–914.

21 KEPLER, T. B., L. CROSBY AND K. T. MORGAN. 2002. Normalization and analysis of DNA microarray data by self-consistency and local regression. *Genome Biol.* **3**:RELEASE 0037.1–12.

22 KERR, M. K., M. MARTIN AND G. A. CHURCHILL. 2000. Analysis of variance for gene expression microarray data. *J. Comput. Biol.* **7**:819–37.

23 KERR, M. K. AND G. A. CHURCHILL. 2001. Statistical design and the analysis of gene expression microarray data. *Genet. Res.* **77**:123–8.

24 KNIGHT, J. 2001. When the chips are down. *Nature* **410**:860–1.

25 LI, C. AND W. H. WONG. 2001. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Porc. Natl Acad. Sci. USA* **98**:31–6.

26 LIPSHUTZ, R., S. FODOR, T. GINGERAS AND D. LOCKHART. 1999. High density synthetic oligonucleotide arrays. *Nat. Genet.* **21 (Suppl. 1)**:20–4.

27 LÖNNSTEDT, I. AND T. P. SPEED. 2002. Replicated microarray data. *Stat. Sinica* **12**:31–46.

28 PICARD, F., S. ROBIN, M. LAVIELLE, C. VAISSE AND J.-J. DAUDIN. 2005. A statistical approach for array CGH data analysis. *BMC Bioinformatics* **6**:27.

29 ROCKE, D. M. AND B. DURBIN. 2001. A model for measurement error for gene expression arrays. *J. Comput. Biol.* **8**:557–69.

30 ROCKE, D. M. AND B. DURBIN. 2003. Approximate variance-stabilizing transformations for gene-expression microarray data. *Bioinformatics* **19**:966–72.

31 SCHADT, E. E., C. LI, B. ELLIS AND W. H. WONG. 2001. Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data. *J. Cell. Biochem.* **Suppl. 37**:120–5.

32 SCHADT, E. E., S. W. EDWARDS, D. GUHATHAKURTA, et al. 2004. A comprehensive transcript index of the human genome generated using microarrays and computational approaches. *Genome Biol.* **5**:R73.

33 SCHUCHHARDT, J., D. BEULE, A. MALIK, E. WOLSKI, H. EICKHOFF, H. LEHRACH AND H. HERZEL. 2000. Normalization strategies for cDNA microarrays. *Nucleic Acids Res.* **28**:e47.

34 SMYTH, G. 2004. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Stat. Appl. Genet. Mol. Biol.* **3**:article 3.

35 SU, A. I., T. WILTSHIRE, S. BATALOV, et al. 2004. A gene atlas of the mouse and human protein-encoding transcriptomes. *Proc. Natl. Acad. Sci. USA* **101**:6062–67.

36 TUSHER, V. G., R. TIBSHIRANI AND G. CHU. 2001. Significance analysis of microarrays applied to the ionizing radiation response. *Proc. Natl Acad. Sci. USA* **98**:5116–21.

37 VON HEYDEBRECK, A., W. HUBER AND R. GENTLEMAN. 2004. Differential Expression with the Bioconductor Project. In SHANKAR SUBRAMANIAM, O. (ed.), *Encyclopedia of Genetics, Genomics, Proteomics and Bioinformatics.* Wiley, New York, NY.

38 WU, Z., R. IRIZARRY, R. GENTLEMAN, F. MARTINEZ MURILLO AND F. SPENCER. 2004. A Model based background adjustement for oligonucleotide expression arrays. *J. Am. Stat. Assoc.* **99**:909–17.

39 WU, Z. AND R. A. IRIZARRY. 2005. Stochastic models inspired by hybridization theory for short oligonucleotide arrays. *J. Comput. Biol.* **12**:882–93.

40 YANG, Y. H., M. J. BUCKLEY, S. DUDOIT AND T. P. SPEED. 2002. Comparison of methods for image analysis on cDNA microarray data. *J. Comput. Graph. Stat.* **11**:108–36.

41 YANG, Y. H., S. DUDOIT, P. LUU AND T. P. SPEED. 2002. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res.* **30**:e15.

42 YANG, Y. H. AND T. P. SPEED. 2002. Design issues for cDNA microarray experiments. *Nat. Rev. Genet.* **3**:579–88.

43 YUE, H., P. S. EASTMAN, B. B. WANG, et al. 2001. An evaluation of the performance of cDNA microarrays for detecting changes in global mRNA expression. *Nucleic Acids Res.* **29**:e41.

44 ZIEN, A., J. FLUCK, R. ZIMMER AND T. LENGAUER. 2003. Microarrays: how many do you need?. *J. Comput. Biol.* **10**:653–67.

26
Classification of Patients

Claudio Lottaz, Dennis Kostka and Rainer Spang

1 Introduction

In microarray gene expression studies tissue samples are examined using microarray chips covering as many as 50 000 transcripts. Automatic classification of patients is a powerful tool for molecular diagnosis as well as for the discovery of novel molecular disease subentities. Exploiting this potential in clinical research is of primary interest and holds great promise. Two early examples for clinical microarray studies are given below.

Roepman and coworkers [89] describe a study on head and neck squamous cell carcinomas. In this disease, treatment strongly depends on the presence of metastases in lymph nodes. However, near the neck, diagnosis of metastases is difficult. More than 50% of patients unnecessarily undergo surgery, while 23% remain under-treated. The authors show that treatment based on microarray prediction is significantly more accurate: in a validation cohort under-treatment was completely avoided, while the rate of unnecessary surgery dropped to 14%. By Alizadeh and coworkers [1] a clinical microarray study on diffuse large B cell lymphomas (DLBCLs) is described. Expression profiles from patients were complemented by profiles from cell lines of known differentiation stage and activation status. The authors were able to identify two different groups of DLBCL patients, characterized by expression profiles similar to germinal center B cells and *in vitro* activated peripheral blood B cells, respectively. Further on, significant differences in the survival rates of the two groups were detected.

From a statistical point of view, the major characteristic of microarray studies is that the number of genes is orders of magnitude larger than the number of patients. For classification as well as class discovery this leads to problems involving overfitting and saturated models. When blindly applying classification algorithms, a model rather adapts to noise in the data than to the molecular characteristics of investigated diseases. Thus, the challenge is to find molecular classification rules and novel disease subclasses that can be generalized from a study cohort to entire disease populations.

Bioinformatics - From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

The most frequent clinical problems addressed by microarray analysis include molecular diagnosis of known disease entities, prognosis of disease outcome and prediction of treatment response. These are all supervised problems, relating expression profiles to additional clinical data. Additionally, the detection of previously unknown molecular subentities of a disease is of great interest. This task is an unsupervised problem, the objective is to detect structure inherent in the expression data without the help of additional clinical information. This chapter is structured accordingly. We start by discussing supervised analysis, covering the general setup of predictive classification and present a selection of classification algorithms. Further on, gene selection and issues of model selection as well as the validation of predictive performance are described. Then we proceed to unsupervised analysis including clustering algorithms, class finding, biclustering, semisupervised approaches and concepts related to the validation of unsupervised analysis results.

2 Molecular Diagnosis

Molecular diagnosis based on gene expression profiles is the most widely used approach in clinical microarray studies. The data consists of gene expression profiles of n patients. In addition, each patient has an attributed class label. The label reflects a clinical phenotype. Phenotypes can include previously defined disease entities, as in the leukemia study of Yeoh and coworkers [117], risk groups, like in the breast cancer studies of van't Veer and coworkers [110] or disease outcome, as in the breast cancer study of West and coworkers [116]. The challenge is to learn expression signatures that allow to predict the correct clinical phenotype for new patients.

It is important that the class labels must not be derived from the expression profiles themselves. This requirement embeds molecular diagnosis into the field of supervised machine learning and defines the difference to unsupervised class finding problems. The latter are discussed in Section 3. There are many more genes on the arrays than patients in the study and gene-to-sample ratios typically are in the hundreds. This is the main difficulty in the supervised approaches. A large number of machine-learning algorithms are available to overcome this problem and in the following we will summarize the basic ideas (see Table 1).

2.1 Problem Statement

We start by presenting a basic framework of supervised machine learning. This enables us to formulate the problem of molecular diagnosis in mathematical terms.

Table 1 Notation of key quantities used throughout the chapter (terms commonly used in the machine-learning literature as well as their counterparts in clinical applications are given).

Symbol	Theory	Application
$F(\boldsymbol{X}, Y)$	data generating distribution	disease population
$(\boldsymbol{X}, Y)$	random variable	new (future) patients
$\{\boldsymbol{x}^{(i)}\}_{i=1}^{n}$	profiles	microarrays from patients in the study
$\{y_i\}_{i=1}^{n}$	labels	clinical phenotypes
n	number of data points	number of patients in the study
p	dimension of input space	number of transcripts on the microarray
K	number of classes	number of phenotypes of interest in the study
$\mathcal{D}$	data set	microarrays hybridized in the study
$c(\boldsymbol{x})$	classifier	diagnostic signature
$c^{\star}(\boldsymbol{x})$	Bayes classifier	best possible signature
L	learning algorithm	method to infer a diagnostic signature
$\mathcal{C}$	function class	candidate signatures
$R[c]$	risk	performance of a signature on the disease population
$\hat{R}[c]$	empirical risk	performance of a signature on patients in the study
$E[c]$	conditional error rate	misclassification rate of a signature
$\bar{E}[L]$	unconditional error rate	misclassification rate of a learning algorithm

2.1.1 Notation

We measure p genes on n patients. The data from each microarray is represented by a *profile* $\boldsymbol{x}^{(i)} \in \mathbb{R}^p$. The corresponding *label*, that encodes one of K clinical phenotypes, is denoted by $y_i \in \mathcal{K} = \{k\}_{k=1}^{K}$.

The profiles are arranged as rows in a matrix $\underline{\mathbf{X}} \in \mathbb{R}^{n \times p}$. All labels together form a vector $\mathbf{y} \in \mathbb{R}^n$. The two quantities $(\underline{\mathbf{X}}, \mathbf{y})$ are called a data set $\mathcal{D}$. It holds all data of a study in pairs of observations $\{(\boldsymbol{x}^{(i)}, y_i)\}_{i=1}^{n}$. Study cases are always samples from a larger disease population. Such a population comprises all patients who had a certain disease, have it now or will have it in the future. Of course one has no access to this population. Nevertheless, it is convenient to make it part of the mathematical formalism. We assume that there is a data-generating distribution $F(\boldsymbol{X}, Y)$ on $\mathbb{R}^p \times \mathcal{K}$. $F(\boldsymbol{X}, Y)$ is the joint distribution of expression profiles and associated clinical phenotypes. The patients who enrolled for the study, as well as new patients who need to be diagnosed in clinical practice, can be modeled as independent samples $\{(\boldsymbol{x}^{(i)}, y_i)\}$ drawn from F. In general, capitalized quantities constitute random variables (population properties) (e.g. $\boldsymbol{X}$ and Y), whereas realizations (study properties) are in lower case (as in $\boldsymbol{x}^{(i)}$ and y_i). We aim for a diagnostic signature with good performance not only on the patients in the study, but also in future clinical practice. In mathematical terms this means that we aim for a well-generalizing *classification rule* $c : \mathbb{R}^p \to \mathcal{K}$.

2.1.2 Loss and Risk

We start with defining a mathematical framework for the performance of a diagnostic signature. In this context we need to distinguish between the performance on the samples in the study and the expected performance in clinical practice. We define a *loss function* l that quantifies the loss of diagnosing profile $\boldsymbol{x}$ to have phenotype $c(\boldsymbol{x})$, given the true phenotype is y:

$$l(\boldsymbol{x}, c(\boldsymbol{x}), y) : \mathbb{R}^p \times \mathcal{K} \times \mathcal{K} \to [0, \infty). \tag{1}$$

A simple loss function is the $0/1$ loss function, which assigns a loss of one to each misclassified sample. Loss functions play a role in signature construction as well as validation and vary from algorithm to algorithm. Let us now define the *risk* of a signature as:

$$R[c] = \mathbb{E}\, l(\boldsymbol{X}, c(\boldsymbol{X}), Y) = \int l(\boldsymbol{x}, c(\boldsymbol{x}), y)\, \mathrm{d}F(\boldsymbol{X}, Y), \tag{2}$$

which measures the expected loss of a diagnostic signature when applied to the entire disease population. It is the performance of the signature in clinical practice. Since we have no access to the population, we do not know F and cannot calculate the risk R explicitly. However, we have access to the patients in the study to approximate R. We define the *empirical risk*:

$$\hat{R}[c] = \int l(\boldsymbol{x}, c(\boldsymbol{x}), y)\, \mathrm{d}\hat{F}(\boldsymbol{X}, Y) = \frac{1}{n} \sum_{i}^{n} l(\boldsymbol{x}^{(i)}, c(\boldsymbol{x}^{(i)}), y_i), \tag{3}$$

where $\hat{F}$ is the empirical distribution function which puts weight $1/n$ on each observed data-point. In the context of a microarray study, the empirical risk for the $0/1$ loss function is the error rate of the signature on patients in the study. It can be easily calculated.

2.1.3 Bayes Classifier and Bayes Error

Before we explain how to build diagnostic signatures in practice we introduce a purely theoretical construct – the best thinkable signature, also called the *Bayes classifier*. While it cannot be built in practice, it is helpful for the development of the theory.

Assume we know what is called the *posterior densities* of F, i.e. $P(Y = k|\boldsymbol{X} = \boldsymbol{x})$. Then the Bayes classifier is defined as:

$$c^\star(\boldsymbol{x}) = \underset{k \in \mathcal{K}}{\operatorname{argmax}}\, P(Y = k|\boldsymbol{X} = \boldsymbol{x}). \tag{4}$$

Its error is called the *Bayes error*. The Bayes error can be different from zero, which means it is impossible to construct a molecular signature that never fails. This is not necessarily a matter of insufficient bioinformatics expertise, but can be an intrinsic property of the disease population F. If the same expression profiles can occur under different clinical phenotypes, it is obvious that false classifications cannot be ruled out entirely. The Bayes classifier can be derived theoretically from the risk defined in Eq. (2): Take the 0/1 loss, i.e. $l(\boldsymbol{x}, c(\boldsymbol{x}), y) = \mathbb{I}_{c(x)\neq y}$, where $\mathbb{I}$ denotes the indicator function. $\mathbb{I}_{c(x)\neq y} = 0$ if $c(\boldsymbol{x}) = y$ and 1 for wrong predictions. Minimizing the risk at some $\boldsymbol{x} \in \mathbb{R}^p$ is equivalent to minimizing the probability of future misclassifications at this point. This, in turn, is the same as always assigning the most probable class. That is precisely the statement of Eq. (4). The Bayes classifier is a purely theoretical construct. The challenge is to approximate it based on study data.

2.1.4 **Minimal Empirical Risk and Maximum Likelihood**

Here, we introduce a general principle for adjusting a signature to a given dataset. We assume a specific dependency structure f of Y on $\boldsymbol{X}$. Often f is a family of functions indexed by parameters. As an example, take the linear model. There one assumes $Y = \boldsymbol{X}^{\mathrm{T}}\beta + \varepsilon$, ε being a random variable. In this case Y depends on $\boldsymbol{X}$ linearly, $f(\boldsymbol{X}) = \boldsymbol{X}^{\mathrm{T}}\beta$ and the parameters are the components β. In general we model $F(Y|\boldsymbol{X}, f)$, i.e. $P(Y|f(\boldsymbol{X}))$, and define the *likelihood* of the observed y_i and $\boldsymbol{x}^{(i)}$ as:

$$P(\mathcal{D}|f) = \prod_{i}^{n} P(Y = y_i|\boldsymbol{X} = \boldsymbol{x}^{(i)}, f)P(\boldsymbol{X} = \boldsymbol{x}^{(i)}). \tag{5}$$

Maximization of the above quantity is the same as taking the log, dropping f-independent terms and minimizing

$$\mathcal{L}_l[f] = -\sum_{i}^{n} \log(P(Y = y_i|\boldsymbol{X} = \boldsymbol{x}^{(i)}, f)). \tag{6}$$

This is equivalent to minimizing the empirical risk defined in Eq. (3) taking the loss function to be $l = -\log P(Y|f(\boldsymbol{X}))$.

2.1.5 **Regularized Risk and Priors**

As mentioned before, the main challenge in microarray-based diagnosis is the large number of genes on the array compared to the few patients in the study. In this section we demonstrate mathematical implications of this situation.

In the previous section we have seen that minimizing the empirical risk can be equivalent to the maximum likelihood approach. This suggests that minimizing the empirical risk over a class of candidate signatures $\mathcal{C}$, i.e. $\hat{c} := \operatorname{argmin}_{c\in\mathcal{C}} \hat{R}[c]$, is a valid approach to molecular diagnosis. Unfortunately, the maximum likelihood approach can lead to ill-posed problems where the optimum is not uniquely determined [108]. For high-dimensional microarray data, this is the case even for simple signature classes. An example is *linear discriminant analysis* (LDA), which will be discussed in Section 2.2.

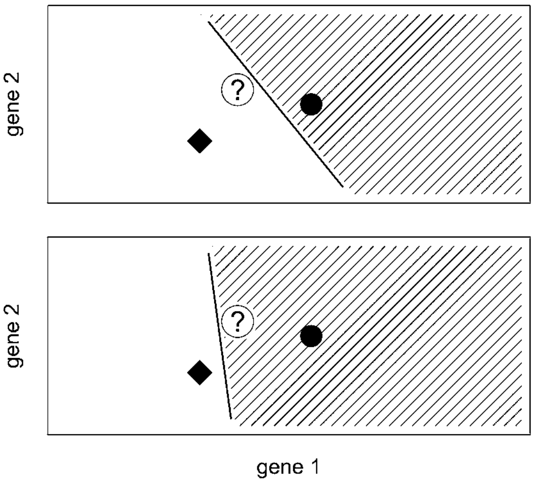


Figure 1 A toy situation with as many genes as patients. The solid diamond and circle represent patients from two different disease types. They can be separated linearly in several ways, two separating signatures are shown. The question mark represents a patient with unknown diagnosis. Both signatures yield conflicting predictions. The data alone does not suggest that one of the signature is better then the other – it does not suggest a unique diagnosis. This situation occurs if the number of genes is higher or equal then the number of patients, which is always the case in microarray studies.

As a simplified illustration imagine a study where two patients, each representative of a certain phenotype, are selected. Then the mRNA abundance of two genes is measured. Further on, we want to construct a linear signature that can discriminate between the two classes. This is the same problem as finding a straight line between two points (each representing a patient) in a plane. The candidate signatures in $\mathcal{C}$ now correspond to all possible straight lines. There is no unique solution (Figure 1). Next, think about a third point which incidentally does not lie on the line going through the first two points. Imagine it represents a new patient with unknown diagnosis. It is always possible to linearly separate the first two points such that the new one lies on the same side as either one of them. The two training patients do not contain sufficient information to diagnose the third patient uniquely. We are in this situation whenever there are more genes than patients. Due to the large number of genes on the chip this problem is inherent in microarray studies.

A way out of the dilemma is *regularization*, which artificially makes the minimizer of the empirical risk unique. In our example above this can correspond to finding the straight line which separates the two points *and* which has maximal distance to both of them. This strategy is implemented in *support vector machines* (SVMs), a classification algorithm discussed in the next section. Regularization results in the minimization of the *regularized risk functional*:

$$\hat{R}_{\text{reg}}[c] := \hat{R}[c] + \lambda\Omega[c], \tag{7}$$

where Ω is called the *regularization operator* and penalizes the complexity of signatures. The parameter λ determines a trade-off between performance on the study data ($\hat{R}$) and complexity of the classifier (Ω).

Introducing a regularization term seems natural from the following perspective. Recall the section about maximum likelihood. There we assumed a dependency f between the two random variables $\boldsymbol{X}$ and Y. If we start modeling prior belief about the dependency f and express them in a *prior distribution* $P(f)$, an application of Bayes rule leads to:

$$\underbrace{P(f|\mathcal{D})}_{\text{posterior}} = \underbrace{P(\mathcal{D}|f)}_{\text{likelihood}} \underbrace{P(f)}_{\text{prior}} \frac{1}{P(\mathcal{D})}, \tag{8}$$

where "posterior" represents the probability of a model f, given the data. Usually f corresponds to a certain diagnostic signature. The $f^\star$ that maximizes the posterior is a reasonable choice to yield a classifier, since it best explains the data at hand and matches prior beliefs. The classifier corresponding to $f^\star$ is called *maximum aposteriori* (MAP) estimate. Finding the MAP estimate is equivalent to minimizing a regularized risk. Recall that the posterior is proportional to the likelihood times the prior. The *likelihood* is, as discussed before, the probability of the data given the model. We have also seen that maximizing the likelihood can be viewed as minimizing a suitable empirical risk. The regularization now comes via the prior. It puts more weight onto models f that we deem more likely than others, without having seen the data. This can resolve ambiguities where the likelihood alone is undecided. More formally we can see the equivalence as follows. Ignore the last factor in Eq. (8), since it does not depend on f. Then recall Eq. (5), set $l = -\log P(Y|f(\boldsymbol{X}))$ and choose $\lambda\Omega[f] = -\log P(f)$ in Eq. (7).

2.2 Supervised Classification

Having introduced basic concepts and terminology of supervised machine learning, we now present a collection of classification algorithms commonly used with microarray data. It is important to distinguish between a diagnostic signature c and a learning algorithm L. A signature takes expression profiles as an input and returns class labels as an output. A learning algorithm is used to build signatures. It takes training data as an input and returns signatures as an output. To review notation, L is a set of rules how to generate a signature $\hat{c}$, given data $\mathcal{D}$ consisting of n independent samples from $F(X, Y)$, i.e. $L : \mathcal{D} \mapsto \hat{c}$. We will discuss gene selection-based methods, penalized logistic regression and SVMs as well as bagging and boosting. Several of these algorithms build linear signatures.

2.2.1 Discriminant Analysis and Feature Selection

We start by modeling the data-generating distribution F in a parametric way: $P(X, Y) = P(\boldsymbol{X} = \boldsymbol{x}|Y = k)P(Y = k)\colon = f_k(\boldsymbol{x})\pi_k$. We further assume that the f_k are Gaussian densities:

$$\begin{aligned} f_k(\boldsymbol{x}) &= |2\pi\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} \exp\big[-\frac{1}{2}(\boldsymbol{x}-\mu_k)^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\mu_k)\big] \quad (9)\\ \log f_k\pi_k &= \log|2\pi\boldsymbol{\Sigma}_k|^{-\frac{1}{2}} - \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}\boldsymbol{x} \\ &\quad +\boldsymbol{x}^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}\mu_k - \frac{1}{2}\mu_k^{\mathrm{T}}\boldsymbol{\Sigma}_k^{-1}\mu_k + \log\pi_k. \quad (10)\end{aligned}$$

For the unknown parameters, unbiased estimates for mean and covariance $(\hat{\mu}_k, \hat{\boldsymbol{\Sigma}}_k)$ can be employed. The priors π_k can be estimated by the relative class sizes $\hat{\pi}_k := n_k/n$. This yields estimates $\hat{f}_k$ and the Bayes classifier can be approximated via $\hat{c}(\boldsymbol{x}) = \operatorname{argmax}_{k\in\mathcal{K}} \hat{f}_k\hat{\pi}_k$. The classification boundary between any two classes i and j is defined in terms of the *log odds*:

$$\eta_{ij}(\boldsymbol{x}) = \log\frac{P(Y = i|\boldsymbol{X} = \boldsymbol{x})}{P(Y = j|\boldsymbol{X} = \boldsymbol{x})} \quad \text{for all } i, j \in \mathcal{K},$$

as the set $\{\boldsymbol{x}|\eta_{ij}(\boldsymbol{x}) = 0\}$ where the posterior probability to belong to either one of the two classes is the same. Assume equal covariance matrices in Equations (9) and (10). This is called the *homoskedastic* case. Then the first two terms in (10) are the same for all classes k and the classification boundaries are linear in $\boldsymbol{x}$. This is called LDA. If the $\boldsymbol{\Sigma}_k$ are assumed different in the two phenotypes (*heteroskedastic case*), the decision boundaries are quadratic. This is then called *quadratic discriminant analysis* (QDA). If the covariances are assumed to be diagonal, i.e. $\boldsymbol{\Sigma}_k = \operatorname{diag}(\sigma_{k1}, \ldots, \sigma_{kp}) =: \operatorname{diag}(\sigma_k)$, one talks about *diagonal discriminant analysis* (DDA). Consequently, the assumption of equal diagonal covariance matrices leads to *diagonal* LDA (DLDA).

Deriving the signature requires the inversion of the estimated covariance matrices. For QDA this leads to the constraints $n_i \geq p + 1$, where the n_i denote the class sizes, and for LDA to $n \geq p + K$. In other words, one needs more patients than genes on the chip, which in our setting is unrealistic. For DDA the estimates $\hat{\boldsymbol{\Sigma}}_k$ are always invertible and it can be applied to expression data directly. With the help of gene selection, LDA and QDA are also applicable to microarray data. Gene selection has to be done prior to the estimation of model parameters. From the entire set of genes only a small number of genes is selected and then discriminant analysis is applied using only the selected subset of genes. Also, the performance of DLDA can be improved using gene selection [30]. A popular method equipping a variant of DLDA with gene selection was proposed by Tibshirani and coworkers [107]. We will discuss it now in some detail.

In the case of LDA, diagnosis consists of classifying a new sample to the class k with the nearest group centroid μ_k (modulo the influence of the priors π_k). Distance is measured in terms of the *Mahalanobis distance* , i.e. $\hat{c}(\boldsymbol{x}) = \text{argmin}_{k\in\mathcal{K}}(\boldsymbol{x} - \hat{\mu}_k)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x} - \hat{\mu}_k)$. Tibshirani and coworkers [107] restrict Σ to be diagonal. Additionally it is pointed out, that the estimate of the mean might be obscured by noise present in the data. Therefore, Tibshirani and coworkers [107] advocate the use of a denoised or *shrunken* centroid for each group k in the distance calculation, i.e.:

$$\begin{aligned} \hat{c}(\boldsymbol{x}) = &\quad \underset{k\in\mathcal{K}}{\text{argmin}}(\boldsymbol{x} - \tilde{\mu}_k)^{\mathrm{T}}\hat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x} - \tilde{\mu}_k) + \log \hat{\pi}_k \\ \tilde{\mu}_k = &\quad \hat{\mu} + \boldsymbol{\Delta}_k(\delta), \end{aligned} \tag{11}$$

where $\hat{\boldsymbol{\Sigma}} = \text{diag}(\hat{\sigma})$ and $\sqrt{\hat{\sigma}_i}$ is the pooled within-class standard deviation of class k for gene i plus a fudge factor s that reduces the effect of nearly constant genes. $\hat{\mu}$ is the usual estimate for the overall mean. The vector $\boldsymbol{\Delta}_k$ extracts only genes in which the mean of group k strongly differs from the overall mean, i.e.:

$$(\boldsymbol{\Delta}_k)_i = \text{sgn}(\hat{\mu}_k - \hat{\mu})_i \left| |(\hat{\mu}_k - \hat{\mu})_i| - m_k(s_i + s)\ \delta \right|_+,$$

where $m_k s_i$ estimates the standard error of $(\hat{\mu}_k - \hat{\mu})_i$ and $|x|_+ = x$ for $x > 0$ and 0 otherwise, i.e. each component of $\hat{\mu}_k$ is shrunken towards the overall mean in units of the standard error. Additionally to the classification function, this approach can yield an estimate of the posterior class probability $P(Y = k|\boldsymbol{X} = \boldsymbol{x})$ by the $\hat{f}_k(\boldsymbol{x})$ and an application of Bayes rule. The value for the gene selection parameter δ is obtained by *cross-validation*, which is discussed in Section 2.4. A link between this approach and classical linear models is discussed by Huang and coworkers [55]. Other flavors of discriminant analysis have been applied to gene expression data as well. For an overview as well as a comparison with other methods, see Refs. [30, 67].

2.2.2 Penalized Logistic Regression

In our discussion above we modeled the data-generating distribution explicitly via the class-conditional probabilities f_k. Here, we take a discriminative approach and model the posterior densities as follows:

$$P(Y = k|\boldsymbol{X} = \boldsymbol{x}) = \frac{\exp[\beta_k^{\mathrm{T}}\boldsymbol{x}]}{\sum_i^K \exp[\beta_i^{\mathrm{T}}\boldsymbol{x}]}, \tag{12}$$

with identifiability constraints. The classification rule corresponding to this model imitates the Bayes classifier: $\hat{c}(\boldsymbol{x}) = \text{argmax}_{k\in\mathcal{K}}\, P(Y = k|\boldsymbol{X} = \boldsymbol{x}, \hat{\beta})$. If we focus on the two-class problem with $y_i \in \{\pm 1\}$ this reduces to $\mathsf{p} := P(Y = 1|\boldsymbol{X} = \boldsymbol{x}) = 1/(1 + \exp(-\beta^{\mathrm{T}}\boldsymbol{x}))$ and $P(Y = -1|\boldsymbol{X} = \boldsymbol{x}) = 1 - \mathsf{p}$.

The likelihood of the class labels can be modeled via independent biased coin flips:

$$\mathcal{L} = \prod_{i=1}^{n} P(Y = y_i | X = \boldsymbol{x}^{(i)}, \beta) = \prod_{i=1}^{n} \mathsf{p}_i^{\frac{1}{2}|y_i+1|} (1 - \mathsf{p}_i)^{\frac{1}{2}|y_i-1|}, \tag{13}$$

and we can find $\hat{\beta}$ as the minimizer of the negative log-likelihood, i.e. $\hat{\beta} = \text{argmin}_{\beta} - \log \mathcal{L}$. The dependence of $\mathcal{L}$ on β is through p and the minimization can be done directly or via an iterated least-squares procedure. Note that this is equivalent to minimizing the empirical risk with a loss function chosen as $l = -\log[1 + \exp(y\beta^{\mathrm{T}}\boldsymbol{x})]$, the *logistic loss function*. Due to the high numbers of genes on the arrays the optimum of Eq. (13) is not unique. As a solution, a regularized risk (see Eq. (7)) can be minimized choosing an appropriate regularization term. Common choices are the L_1 and the L_2 penalty, where $\Omega[c] = \|\beta\|_1^2$ and $\Omega[c] = \|\beta\|_2^2$, respectively. The L_2 penalty is usually combined with gene selection techniques [34, 121], while the L_1 penalty automatically produces sparse solutions, i.e. only few genes contribute to the posterior [91, 95]. For the minimization Roth [91] utilizes an iterative least squares procedure extending an algorithm of Osborne and coworkers [80], while Shevade and coworkers [95] use a Gauss–Seidel method. Kim et al. [62] propose a gradient descent algorithm for problems of this form.

A related approach is that of West and coworkers [116] and Spang and coworkers [100], who model $P(Y = 1|X = \boldsymbol{x}) = \Phi(\boldsymbol{x}^{\mathrm{T}}\beta)$ via the *probit regression model* . Here, Φ denotes the cumulative density function of a standard normal distribution. However, in contrast to what corresponds to a maximum *a posteriori* estimate in the approaches before, a full Bayesian analysis is employed and the posterior distribution of all model parameters is sampled. Regularization is achieved via the introduction of hierarchical normal priors.

2.2.3 **Support Vector Classification**

For simplicity suppose a classification problem with only two possible clinical phenotypes, i.e. $y_i \in \{\pm 1\}$. Then, SVMs [93, 111, 112] fit a maximal (soft) margin hyperplane between the two classes. In high-dimensional problems there are always several perfectly separating hyperplanes (the maximum likelihood approach leads to an ill-posed problem). However, there is only one separating hyperplane with maximal distance to the nearest training points of either class.

This concept is typically combined with the *kernel trick* to allow for flexible nonlinear classification boundaries. The kernel trick is applicable to classification algorithms that can be expressed in terms of inner products of the inputs $\boldsymbol{x}^{(i)}$ who reside in what is called the *input space*. This is the case for the maximum margin hyperplane. The inner products are then substituted by a *kernel function* $k(\boldsymbol{x}, \boldsymbol{x}')$, which corresponds to a *feature map* Φ that maps the profiles from the input space into a *feature space* $\mathcal{H}$:

$$\begin{array}{llll}\Phi: & \mathbb{R}^p & \longrightarrow & \mathcal{H} \\ & x & \mapsto & \Phi(x). \end{array} \tag{14}$$

This results in the original algorithm being now carried out in $\mathcal{H}$ and leads to nonlinear decision boundaries in the input space.

After applying the kernel trick, a class of linear functions in feature space is given by $\mathcal{F} := \{f(x) = \sum_i^n \alpha_i k(x, x^{(i)}) + b \,|\, \alpha_i, b \in \mathbb{R}\}$. The associated diagnostic signatures read $c(x) = \operatorname{sgn} f(x)$. Finding the maximal margin hyperplane is the same as minimizing a regularized risk. The loss function employed is called *soft margin loss* and the regularization term is $\|f\|^2_{\mathcal{H}} := \alpha^{\mathrm{T}}\mathbf{K}\alpha$. Here, $\mathbf{K}$ is the *kernel matrix* and $\mathbf{K}_{ij} = k(x^{(i)}, x^{(j)})$. Regularization is essential to counter the additional flexibility acquired by the kernel trick. In summary:

$$\hat{f}(x) = \underset{f\in\mathcal{F}}{\operatorname{argmin}} \Big\{ \sum_i^n \max(0, 1 - y_i f(x^{(i)})) + \frac{\lambda}{2}\alpha^{\mathrm{T}}\mathbf{K}\alpha \Big\}. \tag{15}$$

In this formulation, separability of the two classes is not required and margin violations are allowed. The trade off between margin violations and margin size (measured as $\|f\|^{-2}_{\mathcal{H}}$) is reflected by the regularization parameter λ. Support vector classification, in combination with various gene selection methods, has been applied to microarray data [42,87] and compared favorably [16, 67, 72]. A SVM-specific *wrapper method* (see Section 2.3.2) for feature selection is presented by Guyon and coworkers [47]. The generalization to more than two classes is not straightforward and different methods are compared by Hsu and coworkers [54].

2.2.4 Bagging

Bagging [14] is a method of aggregating weak classifiers via bootstrapping the data at hand. Bagging stands for **b**ootstrap **agg**regat**ing** and roughly works as follows. M bootstrap samples are drawn from the data $\mathcal{D}$. A *bootstrap sample* from $\mathcal{D}$ is an *iid* sample $\{(x^{(i)*}, y_i^*)\}_{i=1}^{n_{\mathrm{B}}}$ of size n_{B} from the empirical distribution function of the data, which puts weight $1/n$ onto each observation. Then a simple learning algorithm is trained on each bootstrap sample minimizing the empirical risk. This results in a set of weak classifiers $\{\hat{c}_m\}_{m=1}^M$. In the end all weak classifiers $\hat{c}_m$ are averaged to obtain a final strong signature: $\hat{c}(x) = \operatorname{argmax}_k \sum_{m=1}^M \mathbb{I}_{\hat{c}_m(x)=k}$. The class k which most of the $\hat{c}_m$ agree on gets chosen. If the weak classifiers produce estimates of the class conditional probabilities, these can be averaged instead [52].

Random forests [15] constitute an application of this concept. There weak classifiers are derived using classification trees [13] grown using only a random subset of genes. The forest of trees (weak signatures) is then averaged over the bootstrap samples. Random forests are applied to gene expression data, for example, by Gunther and coworkers [46].

2.2.5 Boosting

Another method falling into the category of aggregated classifiers is *boosting*. The algorithm Adaboost was introduced by Freund and coworkers [39]. Several weak signatures c_m are combined to form an aggregate classifier. Hastie and coworkers [52] present Adaboost as an forward stagewise additive modeling approach. The empirical risk is minimized choosing the *exponential loss function* $l(\boldsymbol{x}, c(\boldsymbol{x}), y) = \exp(-yc(\boldsymbol{x}))$ for $y \in \{\pm 1\}$. Basically, the classifier $c(\boldsymbol{x})$ is viewed as an expansion in the weak c_m, i.e. $c(\boldsymbol{x}) = \text{sgn}[\sum_i^M \beta_m c_m(\boldsymbol{x})]$. However, in contrast to bagging, the different c_m are not independently fit to bootstrap samples. Rather, given coefficients $\{\hat{\beta}_i\}_{i=1}^{m-1}$ and classifiers $\{\hat{c}_i\}_{i=1}^{m-1}$, the next pair $(\hat{\beta}_m, \hat{c}_m)$ is determined to optimally supplement the previous ones in terms of minimizing the empirical risk. This results in an iterative optimization strategy. The complexity of the classifiers can be regulated by the number of iterations allowed, i.e. M.

Dettling and coworkers [24] apply this procedure to microarray data. As weak signatures *decision stumps* are used, i.e. classification trees with only two terminal nodes. They use the LogitBoost algorithm of Friedman and coworkers [40]. There, the exponential loss function of Adaboost is exchanged for the logistic loss function introduced earlier. As feature selection, a nonparametric filter method [81] is employed. A combination of bagging and boosting is presented in Dettling and coworkers [25] in the context of gene expression analysis. Both methods are discussed in Tan and coworkers [103]. Zhang and coworkers [119, 120] also apply classification trees in the context of gene expression.

2.3 Gene Selection

Combining classification algorithms with a gene selection procedure is common practice in microarray-based diagnosis. It is done for two reasons. First, and most importantly, gene selection reduces model complexity and in many cases impacts the predictive performance of the signature [30]. Here, model complexity refers to the flexibility of the decision boundaries. Methods like linear discriminant analysis are not applicable at all without gene selection. Other methods, like nearest shrunken centroids [106] or L_1-penalized regression techniques, implicitly reduce the number of genes involved in the signatures. Secondly, the reduction of genes leads to a smaller and hence cheaper design of diagnostic chips or marker panels [57].

Feature selection has a strong impact on the predictive performance of a signature. For this reason it cannot be considered to be a preprocessing step like data normalization discussed in the previous section. It is an essential part of the signature-building algorithm. There is an important difference between building a signature based on 10 genes and building a signature that depends on only 10 genes that, however, were chosen from a pool of 30 000 genes. The first is a low-dimensional model and can be specified by 10 parameters. The second still involves 30 000 parameters, although 29 990 of them were constrained to be zero. The important point is, that it was not agreed which of them should be zero *before* the data was looked at. The importance of not separating gene selection from the signature building process becomes apparent in Section 2.4.

2.3.1 **Filter Approaches**

When feature selection is performed independently of the learning algorithm one talks about a *filter approach* [59]. Straightforward implementations univariately screen for genes, optimizing a score reflecting correlation with the class labels. Popular choices are the t-statistic, the (nonparametric) Wilcoxon rank-sum statistic, the absolute difference of the group means divided by the sum of the estimated standard deviations [45] or the F-statistic in the multiclass case. More sophisticated are filter approaches (heuristically) searching for an optimal *subset* of informative genes [10,50]. Jäger and coworkers [58] and Ding and coworkers [27] also take into account and minimize the redundancy of the selected gene set.

2.3.2 **Wrapper Approaches**

If the learning algorithm is taken into account while looking for informative genes, this is called a *wrapper method* [59]; the feature selection procedure is "wrapped around" the learning algorithm. Either single features or subsets of features are sought that maximize a performance score of the learning algorithm, e.g. the estimated misclassification error. This is a generic procedure and independent of the learning algorithm considered. Looking for optimal feature subsets is a combinatorial problem, and heuristics like forward selection and backward elimination can be employed [59].

A prototypical example is *recursive feature elimination* (RFE) [47]. This approach has been put into the framework of assessing the sensitivity of generalization bounds [85]. This concept, in turn, has been applied to gene expression data by Cho and coworkers [21]. Another approach is to use the penalty term in the regularized risk to ensure sparse solutions (few genes contribute), as it is the case for L_1-penalized logistic regression. Shevade and coworkers [95] use this property together with cross validation (see next section) to simultaneously select and assess the relevance of genes. Such methods are also called *embedded methods* [48].

2.4 Adaptive Model Selection and Validation

This section covers two important steps in microarray-based diagnosis: adaptive model selection and the validation of the predictive performance of a molecular signature. While these are two different tasks, the methodology in use is similar.

2.4.1 Adaptive Model Selection

The algorithms discussed in the previous section can all be linked to minimizing the empirical or the regularized risk (with an appropriate choice of the loss function) over a class $\mathcal{C}$ of candidate signatures. For instance, in the case of penalized logistic regression the signatures take the parametric form of the right-hand side of Eq. (12) and are parameterized by the β_k. For kernel classifiers, $\mathcal{C}$ corresponds to signatures that can be written in the form $f(\boldsymbol{x}) = \text{sgn}[\sum_i^n \alpha_i k(\boldsymbol{x}, \boldsymbol{x}^{(i)}) + b]$ with α_i and $b \in \mathbb{R}$. Depending on how rich this class of signatures is, the learning algorithm is able to implement more or less flexible boundaries between the K phenotypes. For microarray data, the typical situation is that even simple signature classes, such as hyperplanes, are extremely rich due to the high dimensionality of the profiles $\boldsymbol{x}^{(i)}$. For simplicity, we will refer to the richness of the set of candidate signatures as the complexity of a *diagnostic model*.

2.4.1.1 **Bias-variance Trade-off** When dealing with complex diagnostic models not the empirical risk needs to be optimized, but the regularized risk in Eq. (7). This allows for controlling complexity. The regularization term introduces a complexity penalty, thus effectively restricting the complexity of the derived diagnostic signature $\hat{c}$. The regularization parameter λ quantifies the trade-off between model fit and model complexity. With little regularization the algorithm can fit very flexible decision boundaries to the data. This results in few misclassifications on the data from patients in the study. Nevertheless, it can have poor predictive performance in clinical practice. The reason is that the algorithm not only fits population properties (as desired), but also reflects noise resulting from patient sampling. We refer to this as *overfitting*. When the regularization term dominates Eq. (7), the resulting signatures might be too restricted. Then we have poor performance on both, the study patients and in future clinical practice. We refer to this situation as *underfitting*. The problem described above is also known as the *bias-variance trade-off*, since regularization introduces a bias into the estimation of model parameters, while at the same time reducing the sample variance. Sample variance here refers to $\hat{c}$ varying

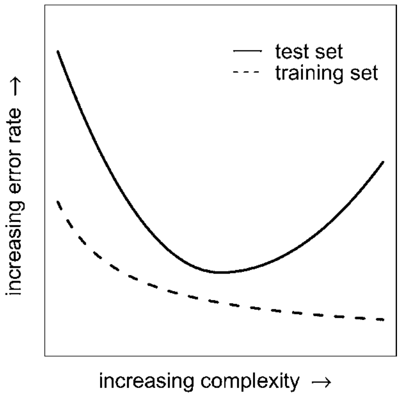


Figure 2 Schematic diagram to illustrate the *bias-variance trade-off* (see text). The x-axis codes for model complexity and the y-axis for error rates. The dashed line displays the training error, the solid line the test error. Low-complexity models produce high test errors (underfitting, low variance, high bias) and so do highly complex models (overfitting, high variance, low bias).

with different cohorts drawn from the same disease population. See also Figure 2.

2.4.1.2 **Choosing a Trade-off via the Hold Out** The regularized risk is a general example of how learning algorithms can implement a tuning parameter (in this case λ) that allows for balancing over- and underfitting. Further instances are the number of genes included after the variable selection process or the amount of shrinkage in the nearest shrunken centroid method.

Before we start explaining how these parameters can be tuned, we need some more notations: recall that we need to build signatures from a finite data set $\mathcal{D}$, drawn from F. Let us define the *empirical error rate* of a signature $\hat{c}$ built on $\mathcal{D}$ as:

$$\hat{e}_{\lambda}^{\text{emp}}\left[\,\hat{c}\,\right] := \frac{1}{n}\sum_{i}^{n}\mathbb{I}_{\hat{c}_{\lambda}(x^{(i)})\neq y_i}, \tag{16}$$

where we have made the dependency of the classification rule on the regularization parameter λ explicit and $\mathbb{I}$ denotes the indicator function. The empirical error rate is equivalent to the empirical risk $\hat{R}$ when using the $0/1$ loss function. The empirical error rate is a random variable since it depends on the random sampling of patients that were included into a study. When repeating the study with a second cohort of patients, one obtains a different empirical error rate.

We now split the data set into a *training* or *learning set* $\mathcal{D}_l$ and a *test set* $\mathcal{D}_t$. The training set constitutes a (smaller) study cohort, while the test set can be used like novel patients with unknown diagnosis. In this respect, we want little errors on the test set. To reduce test errors, we are even willing to pay a price in terms of some more errors on the training set. For a fixed value of the regularization parameter, one can learn a signature $\hat{c}_\lambda(\boldsymbol{x}) := L(\boldsymbol{x}; \mathcal{D}_l)$ by applying a learning algorithm only to the training data. Subsequently, its predictive performance can be evaluated by applying the generated signature to the independent test data and calculating the error on the test set only. The value of the regularization parameter can be varied, identifying a value such that the above estimate is minimal. That is, the learning algorithm then consists of minimizing $\hat{R}_{\text{reg}}$ and choosing the regularization parameter via assessment of generalization performance. This procedure is referred to as *adaptive model selection*, since by determining the regularization parameter one chooses a model with approximate optimal complexity. However, different clinical classification problems need different amounts of regularization. By using a hold out set, model selection is adapted to the data at hand.

2.4.1.3 **Using Data More Efficiently via Cross-Validation** Since microarray data is expensive and scarce, one can make use of the following procedure. The data set $\mathcal{D}$ is randomly partitioned into Q bins $\{\mathcal{D}_q\}_{q=1}^{Q}$. Each one of the $\mathcal{D}_q$ is then used as hold out set in turn. More formally: Let $\kappa : \{1, \ldots, n\} \rightarrow \{1, \ldots, Q\}$ be a partitioning, i.e. $\kappa(i) = q$ for all $i \in \mathcal{D}_q$, $i \in \{1, \ldots, n\}$ and $q \in \{1, \ldots, Q\}$. Further, let $\hat{c}_\lambda^{-\kappa(i)} := L(\boldsymbol{x}; \mathcal{D} \setminus \mathcal{D}_{\kappa(i)})$ be a classifier trained on $\mathcal{D} \setminus \mathcal{D}_{\kappa(i)}$ for a fixed value for λ. Then we estimate the misclassification rate via [43,102]:

$$\hat{e}_\lambda^{cvq}\,[\,\hat{c}\,] := \frac{1}{n} \sum_{i}^{n} \mathbb{I}_{\hat{c}_\lambda^{-\kappa(i)}(\boldsymbol{x}^{(i)}) \neq y_i}. \tag{17}$$

This quantity estimates the expected error rate on future data. Again, one can do this for a grid of values of the regularization parameter and an approximately optimal value can be identified. Adaptive model selection is a part of the learning algorithm, as it was the case with gene selection. This needs to be kept in mind when assessing the performance of a signature.

2.4.2 Validation of the Predictive Performance of a Molecular Signature

After having derived a diagnostic signature one needs to estimate its expected performance in future clinical practice. This validation step constitutes one of the most critical steps in the whole process of molecular diagnosis and several pitfalls are involved. Estimators can be overly optimistic (biased) or they might have high sample variances. It also makes a difference whether one is interested in estimating the performance of a fixed signature $\hat{c}$ (which is usually the case in clinical studies) or if one is interested in estimating the performance of the learning algorithm L that builds the signatures (which is usually the case in methodological projects). The performance of the fixed signature $\hat{c}$ varies due to the random sampling of the test set, while the performance of the algorithm L varies due to sampling of both the training and test set.

The two different situations correspond to two different theoretical error rates. The performance of a fixed signature $\hat{c}(\boldsymbol{x}) =: L(\boldsymbol{x};\mathcal{D})$, derived from a training set $\mathcal{D}$, is measured by the *conditional error rate(s)* or the *true error*:

$$\begin{aligned} E_{ij}[\,\hat{c}\,] &= \; P(L(\boldsymbol{X};\mathcal{D}) = j|Y = i,\mathcal{D}) \quad i \neq j \in \mathcal{K} \quad \text{and} \\ E[\,\hat{c}\,] &= \; P(L(\boldsymbol{X};\mathcal{D}) \neq Y|\mathcal{D}). \end{aligned} \tag{18}$$

The first quantity, E_{ij}, is the probability that the signature $\hat{c}$ will classify a patient from the disease population to belong to class j even though they actually belong to the phenotypical class i. The second quantity only asks for wrong classifications of $\hat{c}$, no matter which group is mistaken for what other group. These quantities are not obtainable in practice, since the probabilities need to be calculated with respect to the unknown population distribution $F(\boldsymbol{X},Y)$.

If, in contrast, one is interested in the performance of the learning algorithm L, the sampling variability of the training set has to be taken into account. This makes the conditional error rates random variables. Keeping the size of the training set fixed and taking expectations leads to the *(unconditional) error rate(s)* or the *expected error* :

$$\begin{aligned} \bar{E}_{ij}[\,L\,] &= \mathbb{E}_{\mathcal{D}}\,E_{ij} = P(L(\boldsymbol{X};\mathcal{D}) = i|Y = j) \quad i \neq j \in \mathcal{K} \quad \text{and} \\ \bar{E}[\,L\,] &= \mathbb{E}_{\mathcal{D}}\,E \;\; = P(L(\boldsymbol{X};\mathcal{D}) \neq Y). \end{aligned} \tag{19}$$

As it was the case for the conditional error rates, these quantities depend on F and are not accessible. Hence, both rates need to be estimated using the data at hand.

2.4.2.1 **Estimating Error Rates** One might assume the empirical error [Eq. (16)] can be employed to estimate the conditional error rate. The main problem with this approach is that it uses the *same* data in $\mathcal{D}$ to train the classifier and to evaluate it later on. This can result in highly biased error rates grossly underestimating the true error. This is practically relevant in gene expression data analysis, since the high dimensionality of the data makes algorithms without complexity control prone to overfitting [3,96].

A better approach is to use an independent test set. Only training data is used for gene selection, classifier learning and adaptive model selection. The final signature $\hat{c}$ is then evaluated on an independent test set. Unfortunately, this estimator can have a substantial sample variance, due to the random selection of patients in the test set. This is especially the case if the test set is small. It falls in this line of thought that good performance in small studies can be a chance artifact [79].

More effective use of the data at hand can be made via the cross-validation procedure introduced earlier. The leave-one-out version produces an estimator of the unconditional error rate with almost no bias. It is computationally more expensive than q-fold cross-validation and suffers from a very high sample variance. The latter is reduced for moderate q such as "somewhere between 5 and 10" [52, 64, 73]. Braga-Neto and coworkers [12] advise to average over many different partitionings. No unbiased estimator of the variance of the cross-validation estimate, i.e. valid for all distributions F, exists [7]. Cross-validation error rates naturally refer to the classification algorithm L. In each iteration a different classifier is learned, based on (somewhat) different training data. The cross-validation performance is the average of the performance of different signatures. Nevertheless, cross-validation performance can be also used as a bias-corrected estimator of the conditional error rate. In fact, in applied work it is often used to validate fixed signatures that were derived by the evaluated algorithm.

Efron and coworkers [33] apply *bootstrap smoothing* to the leave-one-out cross-validation estimate. The basic idea is to generate different *bootstrap replicates* $\{\mathcal{D}_b^*\}_{b=1}^{n_B}$, apply leave-one-out cross-validation to each and then to average the results. A result of this approach is the so called "0.632+ estimator". It takes into account the possibility of overfitting and reduces the variance compared to the regular cross-validation estimates. Ambroise and coworkers [3] have found it to work well with gene expression data.

2.4.2.2 **Selection Bias and Nested Loop Cross-validation** As we have discussed in the previous section, feature selection techniques are a central element in the analysis of microarray data. In filter approaches, special care has to be taken when using cross-validation: *the feature selection is part of the learning algorithm L*. For this reason feature selection has to be repeated again on each $\mathcal{D} \setminus \mathcal{D}_q$, i.e. Q times. Global gene selection before the cross-validation (which is also called *incomplete cross-validation* or *information leak*) can result in grossly over-optimistic (biased) estimates of the error rates [3]. For example, Simon and coworkers [96] describe a case, where the incomplete cross-validation method and the fully cross-validated method result in estimated error rates of 27% and 41%, respectively. Similarly, assume the algorithm L contains an adaptive model selection procedure. To get reasonable error rate estimates via cross-validation, the selection procedure has to be applied to every $\mathcal{D} \setminus \mathcal{D}_q$ separately. This leads to a cross-validation step inside a cross-validation, i.e. to a *nested loop* cross-validation [43, 92]. Applying the selection procedure to the complete data can lead to biased estimation of the error and over optimistic results. Ruschaupt and coworkers [92] and Wessels and coworkers [115] realize such a *complete validation procedure* and compare various methods.

Ntzani and coworkers [79] and Michiels and coworkers [75] report that, at least in studies up to 2003, many of 84 considered studies lacked appropriate validation of derived signatures. In fact, many of the shortcomings could have been avoided keeping in mind the two points above.

2.5 Discussion

In the previous sections we introduced basic concepts and methods used in classification problems. Whenever possible, we tried to point out aspects specific to the classification of microarray data. General literature about machine learning includes Refs. [26, 29, 52, 88, 93], where a more thorough treatment of the theoretical concepts can be found. More focused on the analysis of microarray data are Refs. [73, 101].

The methodology above was presented in the classification context only. One might be tempted to interpret the genes driving the models, but this is dangerous. First, it is unclear how the regularization term biases the selection of signature genes. While a bias is a blessing from the diagnostic perspective, this is not the case from the biological perspective. Second, signatures are generally not unique: While outcome prediction for breast cancer patients has been successful in various studies, e.g. [86, 99, 109], the respective signatures do not overlap at all. Further on, Ein-Dor and coworkers [35] derived a large number of almost equally performing signatures in a single dataset. This is not too surprising considering the following: the molecular cause of a clinical phenotype might involve only a small set of genes. This primary event has secondary influences on other genes, which in turn deregulate more genes and so on. In clinical microarray analysis we typically observe an avalanche of secondary or later effects, often involving thousands of differentially expressed genes. While complicating biological interpretation of signatures, such an effect does not compromise the clinical usefulness of predictors. On the contrary, it is conceivable that only signals enhanced through propagation lead to a well-generalizing signature.

3 Finding Molecular Disease Entities

In the previous section we were concerned with reconstructing phenotypically defined disease entities based on expression profiles. This is a supervised learning problem and results in diagnostic signatures with obvious clinical relevance. In this section we go beyond the level of already established clinical phenotypes and aim to define novel disease entities exclusively based on the molecular properties that are reflected in expression profiles. This is an unsupervised learning problem and results in novel patient stratifications. The clinical relevance of such results must always be proved in subsequent analysis. For instance, clinical follow-up studies may confirm that patients belonging to different molecular entities respond differently to treatment. If this is the case, treatment can be adjusted according to this novel molecular diagnosis.

In various microarray studies researchers have applied unsupervised machine-learning techniques to discover molecular disease subgroups, and subsequently validated the clinical impact of the stratification. For instance, patient subgroups with particular outcome probabilities were characterized in Ref. [8], and so far unclear disease mechanisms have been uncovered in Refs. [1,76].

3.1 Clustering

The most widely used approach to molecular disease characterization is via clustering algorithms. Although this approach suffers from several limitations, which we will point out below, we cover the most frequently used methods here. Clustering algorithms are discussed in detail in Chapter 27 in the context of clustering genes. In this section, we focus on specific issues when applying the same algorithms to the clustering of patients.

3.1.1 Clustering Algorithms

Hierarchical clustering is a well-studied and established method in the statistical community, and is by far the most widely used method for clustering patients. It was first applied to microarray data in 1998 by Eisen and coworkers [36]. Since then it has been applied in many clinical microarray studies to support distinctions between patient groups, which are claimed to be coherent from a molecular point of view. Clinical fields of these studies include lung cancer [8], lymphomas [1,76], leukemia [20,45,117], breast cancer [82], ovarian cancer [114], cutaneous melanoma [9], colon cancer [2], glioma [38, 41] and parathyroid tumors [53]. The result of hierarchical clustering is typically illustrated using dendrograms (Figure 3), where similar samples are depicted close to each other. Cutting dendrograms at any level naturally defines a set of clusters, but is subject to arbitrary decisions on where to cut.

Other clustering methods also based on distances between samples directly aim to separate the set of samples into a given number of clusters. Examples of such methods are *k*-means clustering [51], partitioning around medoids [60] and self-organizing maps [65]. These methods, however, are restricted to detect a predefined number of clusters. The FOREL method, otherwise very similar to *k*-means, overcomes this limitation by determining one cluster at a

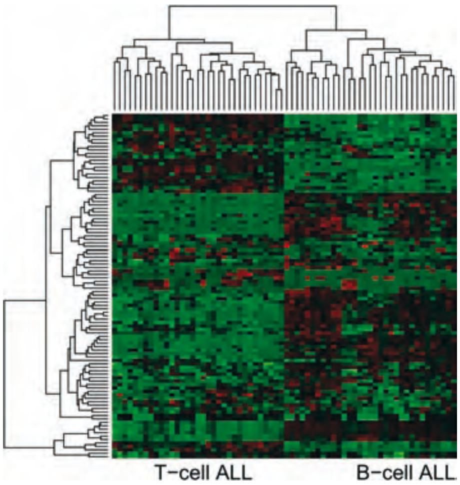


Figure 3 Dendrograms illustrate results of hierarchical clustering. The cluster plot shown is generated from microarray data on acute lymphocytic leukemia (ALL) partially published in Ref. [20]. Half of the samples chosen are from T cell ALL, the other half from B cell ALL. The dendrogram at the top of the image shows that hierarchical clustering clearly separates these two classes.

time, removing its samples and then restarting to find the next cluster [83]. Alternatively, methods for automatic determination of the most adequate number of clusters are suggested in Refs. [31, 66, 105].

In Ref. [19], probabilistic clustering based on a mixture model is suggested. The authors present an algorithm to determine model parameters by Bayesian inference. A similar approach is implemented in mclust [37], a freely available add-on package for the statistical computing environment R [56, 84]. It was applied to microarray data in Ref. [118]. Probabilistic model-based clustering is discussed in further detail in Ref. [98] together with a suggestion of how to choose model parameters in a cross-validation setting.

3.1.2 **The Problem of Distances**

All of the clustering methods described above compute distances between objects, which are in this case patients. Typical choices are the Euclidean distance or a distance based on Pearson's correlation coefficient. Each patient is described by a long list of gene expression values, but not all of them can be assumed to carry information on disease states. Hence, it makes no sense to include all genes in the distance function. Instead, clusters are computed based on a subset of the genes measured in the expression profiles. Including different sets of genes into a Euclidean metric, leads to different distances between patients and consequently influences the results of clusterings. Nevertheless, there is no such thing as a justifiable best choice of features.

The most straightforward selection method is simple selection for maximal variance or maximal bandwidth of expression values within a gene (e.g. Ref. [18]). The motivation for this is to avoid measurement noise expected to be dominant in the many low variance genes. Bhattacharjee and coworkers [8] as well as Monti and coworkers [76] suggest to select reproducibly variable genes. In order to do so they extract two samples from a number of tumors and determine a robust *F*-statistic to capture the reproducibility of genes together with their corresponding variance. Genes are then selected for low variance between replicates and high variance between patients.

3.2 Searching for Partitionings

3.2.1 Overlapping Partitionings

Clustering of patients returns the dominating structure of the patient space. However, from a clinical perspective we expect that different sets of genes uncover different structures on patients, which can all be relevant in their own right. In the context of oncology we can think of one set of genes that partitions the patients with respect to the regulation of the apoptosis pathway, while a second set of genes classifies patients according to the proliferation rate of tumor cells.

Systematically combining clustering with variable selection is one feasible approach to uncover multiple structures in the patient space. For instance, Dougas and coworkers [32] suggest an iterative exclusion of cluster supporting genes in order to find new clusterings in each iteration. In each step, 2-means clustering determines one clustering of the data and the most differential gene with respect to this clustering is removed. During this procedure, various differing clusterings are discovered.

3.2.2 Search and Find

The common aspect of algorithms discussed here is that they search in the space of all partitionings for interesting splits. Different criteria to characterize interesting partitionings are suggested and local or global search algorithms to find them are applied. When searching for homogeneous groups in patients, a desirable property is good separability. This property inspires a group of unsupervised methods, which search in the space of possible partitionings by optimizing a score for separability.

3.2.3 ISIS – Identifying Splits with Clear Separation

The ISIS method [113] is restricted to detecting bipartitions or splits of the data. To this end it uses the diagonal linear discriminant score (DLD) to measure separability of the two classes in a split. For computing the DLD scores, all samples are projected on one dimension as in DLD analysis (DLDA) to discriminate between the two candidate classes. The DLD score is then the two-sample *t*-statistic for the projected values. ISIS detects multiple structures by searching for splits with local minima of the DLD score. Figure 4 illustrates the result obtained from a single call to this method. In Ref. [61], the ISIS method is applied to patients with congenital heart defects. The authors of this study were able to confirm and further characterize molecular subclasses of this disease.

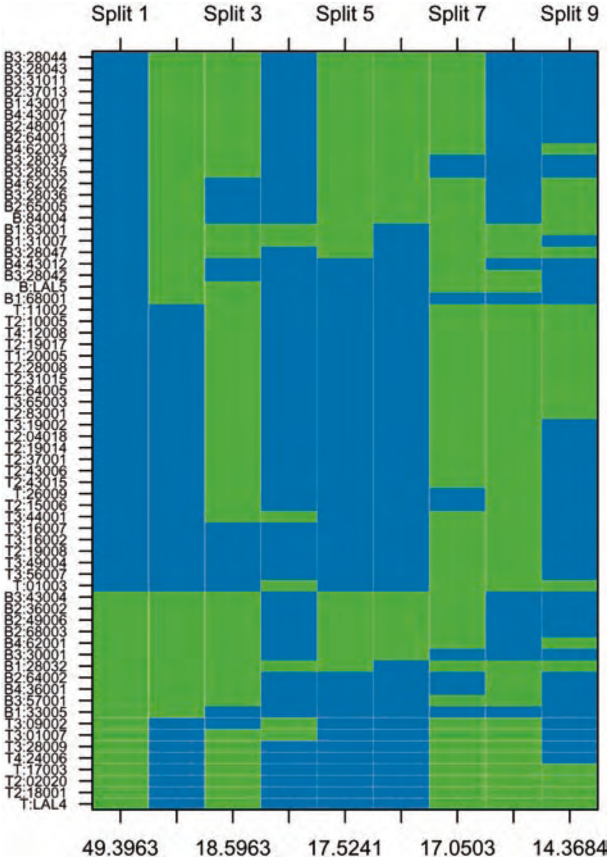


Figure 4 The ISIS method reports a set of locally optimal bipartitionings. Here, it was applied on a subset of samples from the leukemia dataset in Ref. [20] including B cell and T cell ALL cases. The 500 most variable genes are used to compute the results shown here. Each row in the figure corresponds to one of the samples, each column to a split suggested by ISIS. The colors indicate the partition of the cases and the scores on the bottom are DLD scores. Split 2 in the figure corresponds to the separation of T and B cell ALL.

3.2.4 Overabundance of Differential Genes

An alternative approach to class finding is described in Ref. [6]. The authors define a figure of merit measuring the overabundance of genes supporting one split compared to a null model for random splits. This provides a notion of significance for candidate splits. Splits with optimal figure of merit are searched using simulated annealing. This probabilistic process is iterated to yield several alternative splits. An extension from bipartitions to multiple partitions is described in Ref. [68].

3.2.5 Best-fitting Gaussian Model

The two methods described above are heuristics. A full statistical model for class discovery has been described in Ref. [90]. A two-class Gaussian mixture model is used for finding patient splits. In addition, the authors modify the well-known expectation-maximization (EM) algorithm to incorporate feature selection in the M-step using L_1 regularization. Finally, a stability criterion using artificial noisy versions of the data is used to test the robustness of partitionings. The method reports only partitionings which prove stable in the resampling step.

3.3 Biclustering

Methods discussed so far mostly insist on partitioning the entire set of patients or samples. In contrast, biclustering algorithms directly search for homogeneous subgroups of patients which may be based on a subsample of genes. Thus, there are opportunities to search more freely for homogeneous groups of patients, while leaving out the samples which are difficult to associate with others. An overview on suggested algorithms is given in Ref. [71].

Researchers have suggested various versions of biclustering algorithms with differing applications in mind. Getz and coworkers [44] rediscovered the distinction between lymphocytic and myeloid leukemia in the dataset in Ref. [45] in a completely unsupervised manner. In the same analysis, the authors report the rediscovery of the distinction between T and B cell ALL as well as the detection whether patients underwent treatment. For each of these distinctions supporting genes are determined. In a dataset on colon cancer by Alon and coworkers [2], biclustering detected a change in protocol for the data acquisition. Similar proofs of principle are reported in Refs. [63,78,94,104] on data from lymphoma, leukemia, breast cancer, multiple sclerosis and central nervous system embryonal tumors.

3.4 Semisupervised Methods

So far we have discussed both supervised and unsupervised analysis of patient profiles. Recently, several papers have described analysis strategies, which carry characteristics of both fields in parallel. We will call them semisupervised methods.

A major problem with unsupervised methods is that they detect any structure among patients. Novel stratifications can be of clinical relevance, but in many cases they just reflect trivial differences like age and gender. Even worse, clusterings may be detected due to experimental artifacts or noise. The idea of semisupervised methods is to direct class discovery towards clinically relevant partitionings through additional phenotypical information on the patients.

3.4.1 Molecular Symptoms

From the hypothesis that known clinical phenotypes may be caused through different molecular causes, Lottaz and coworkers [70] have derived a semisupervised approach called structured analysis of microarray data (StAM). As input the method needs both expression profiles and class labels reflecting a relevant clinical phenotype, typically a disease group versus a control group. Unlike purely supervised methods StAM does not aim to find a single molecular signature, but searches for hidden subentities (the unsupervised aspect), which need to be subsets of the predefined disease group (the supervised aspect). Then it searches for molecular characteristics, which distinguish the newly found subgroup of disease patients from all other patients. These characteristics are called molecular symptoms in contrast to the molecular signatures derived from purely supervised analysis. They are similar to clinical symptoms, since these as well are not necessarily present in all patients, but almost never occur in healthy people. The method searches for molecular symptoms using classifiers based on biologically focused sets of genes. Candidate sets are deduced from functional annotations collected in the Gene Ontology (GO) [4]. Figure 5 shows the structured analyses concerned with mixed lineage leukemias from the leukemia dataset by Yeoh and coworkers [117].

3.4.2 Survival-driven Class-finding

Outcome prediction is a particularly important issue in clinical studies. Most methods described in the literature combine cluster analysis and classification methods. In this respect, they can be considered semisupervised.

One approach to find classes related to survival is described in Bullinger and coworkers [17]. In a first step, a cox proportional-hazards model is

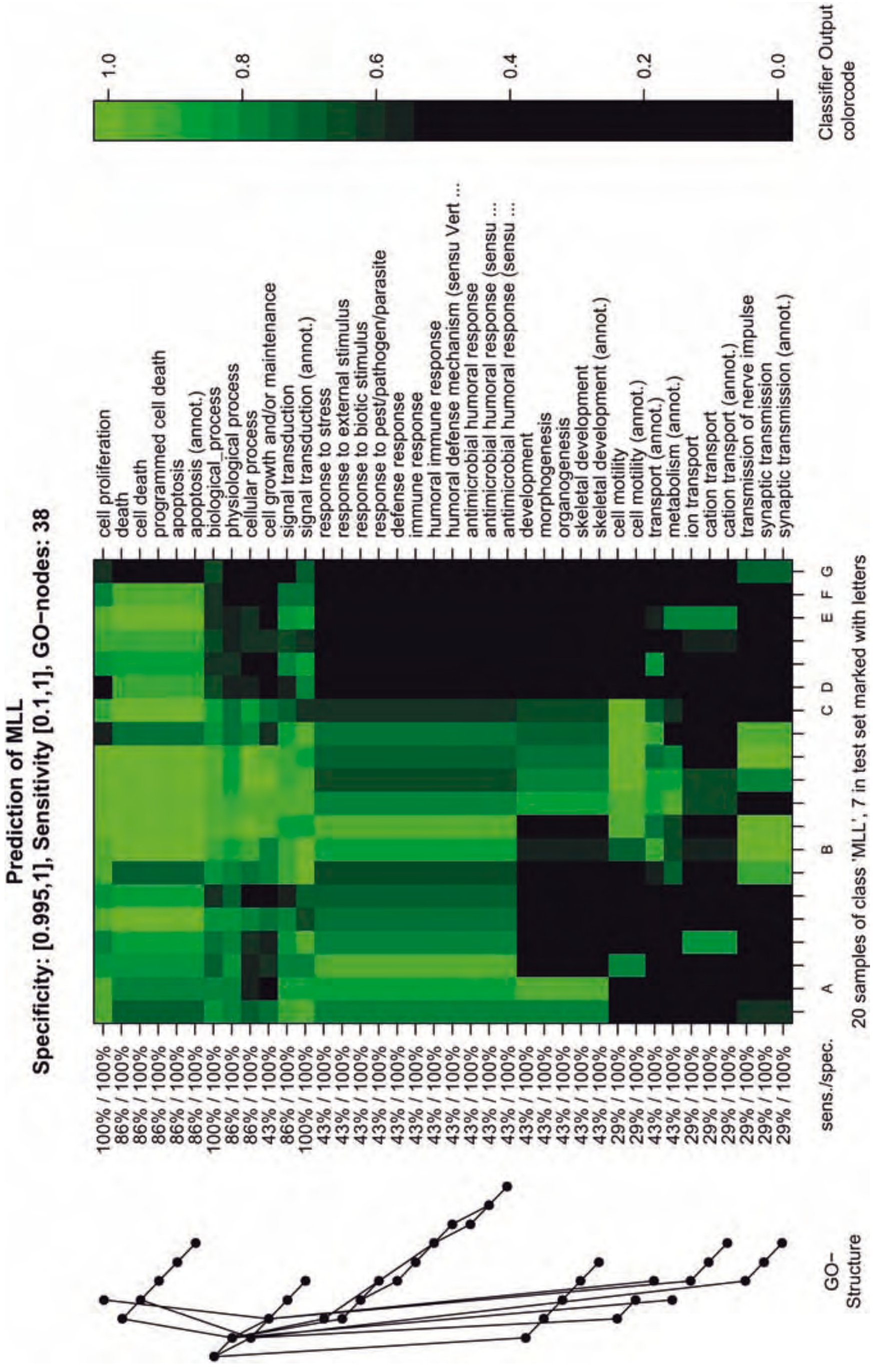


Figure 5 Molecular symptoms stratify patients. Rows correspond to GO-node-based classifiers, columns represent patients with mixed-lineage leukemia. Colors in the image encode classifier results. Bright regions represent presence and dark regions represent absence of molecular symptoms. Presence/absence patterns of molecular symptoms provide a molecular patient stratification.

used together with a permutation test to find genes, which are significantly correlated to outcome. k-means clustering is used on the expression data restricted to these outcome-related genes to separate patients into two groups. Finally, Bullinger and coworkers use the nearest shrunken centroid classification to confirm the separability of the two groups in a cross-validation scheme. The authors have evaluated their method on data from a clinical study on acute myelogenous leukemia in adults where they molecularly characterize a separation into good- and poor-outcome patient groups.

3.4.3 Towards Survival Prediction

Dave and coworkers [22] suggest to determine survival correlated genes similar to Ref. [17], but organize them into signatures associated with poor prognosis and signatures associated with good prognosis by hierarchical clustering on genes. A linear combination of these signatures was used to compute survival scores used to separate patients into four different classes of expected survival. The authors validate their approach on data from a clinical study on follicular lymphoma. A similar approach more focused to determine a small, clinically practical signature for survival prediction of diffuse large B cell lymphoma patients is described in Ref. [69]. Its validity is confirmed on data from independent clinical studies.

3.5 Validating Unsupervised Analysis

Compared to supervised analysis, the validation of clustering and class-finding results is notoriously difficult, since predictive performance cannot be assessed. Researchers have developed various quality indices based on significance of structure, stability of clusters and over-representation of pathways in the genes defining the classes.

3.5.1 Statistical Significance

In order to estimate statistical significance of a clustering, Monte Carlo methods are suggested to estimate the density distributions of indices under the null hypothesis of random and unstructured data [49, 74]. Bolshakova and coworkers [11] provide a tool to evaluate clusterings and quality indices interactively.

3.5.2 Stability

Lange and coworkers [66] suggest to measure cluster stability computed as follows. The data at hand is split many times into training and test set. For each split, the clustering algorithm is applied to both data sets, yielding labels for all samples. The training sets are then used to train a classifier, the corresponding test sets are used to determine misclassification rates compared to the labels given by the clustering algorithm. Lange and coworkers suggest to use average misclassification rates to measure stability of clusterings, expecting low values for stable clusterings.

3.5.3 **Detect Consensus by Subsampling**

Consensus matrices computed on subsampled data are suggested in Ref. [77] to validate cluster stability. These are computed as follows. N random equal size subsets of samples are generated. For each of them the same clustering algorithm is applied to compute N perturbed partitioning. Each column and each row of a consensus matrix represent one patient, rows and columns are in the same order, such that elements from the same cluster in the partitioning of the complete data are adjacent. The element in row i and column j of the consensus matrix contains the relative frequency that patients i and j fall in the same cluster. For a stable clustering, the consensus matrix holds values close to zero and close to one. In this case, the consensus matrix holds square blocks along the diagonal. Figure 6 shows a consensus matrix from the leukemia data in Ref. [20] with two stable clusters (upper panel) and a consensus matrix computed from the same data, but restricted to only B cell samples, which displays a less apparent structure (lower panel). Monti and coworkers [77] suggest to use average consensus values per cluster to identify stable clusters as clusters with consensus value close to 1. Similarly, they use average consensus values per sample to estimate if it is clearly attributed to its cluster. Consensus clustering revealed three stable clusters in more than 200 diffuse large B cell lymphoma patients [76].

3.5.4 **Adding Simulated Noise**

An alternative to subsampling in order to perturb the clusterings is to add noise from a Gaussian distribution to all expression levels and evaluate the stability of clusters with respect to this noise. This is suggested in Ref. [74].

3.5.5 **Over-represented Pathways**

Evidence for the biological relevance of a molecular patient entity is given when the list of driving genes has a clear biological focus. Therefore, biologists often investigate functional annotations of these genes, e.g. using the GO. Statistically significant over-representation of certain pathways or biological processes in the list of genes that define a molecular partitioning of patients can be used as validation for the partitioning itself. Over-representation analysis is described in Refs. [5, 23, 28]. It is used as argument for cluster validity in Ref. [76].

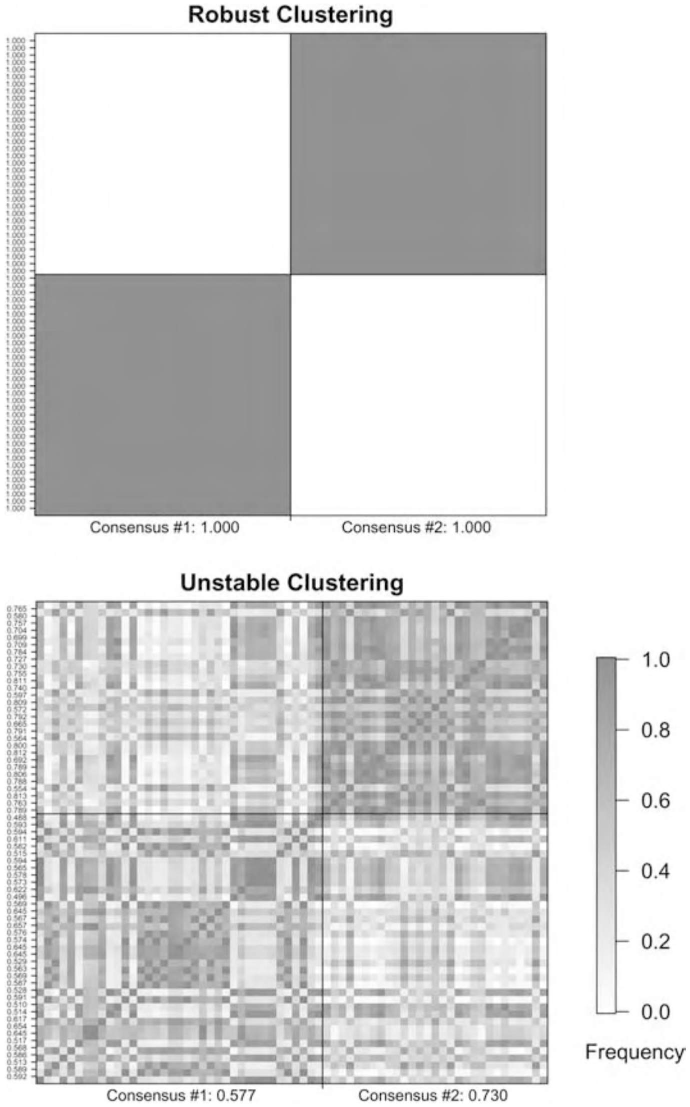


Figure 6 Consensus matrices for subsets of leukemia patients from Ref. [20]. The set of patients used to generate the upper consensus matrix contains T and B cell leukemias. Expression profiles of these differ strongly and yield a clear consensus matrix. To generate the consensus matrix in the lower panel, only B cell leukemias were used and no obvious separation was found.

4 Conclusions

Both computational diagnostics and clinical class discovery using microarray gene expression profiling are established fields in clinical research. Currently, however, the importance of microarray studies is limited to basic research, while there is little impact on therapeutical decision making. Some reasons for this are discussed in Ref. [97].

Important obstacles for large enough and sufficiently focused microarray studies include stringent planning of unified protocols and procedures, acquisition and selection of patient material, appropriate logistics as well as financial support. The need for many patients to conduct a decent clinical study often implies that material from several hospitals must be used. Such complex collaborations, however, enforce meticulous planning to ensure homogeneous procedures as well as efficient logistics to make sure that probes are handled quickly and reliably. Finally, such studies are expensive in labor and material, given that several hundreds of patients are to be enrolled to provide clinically relevant conclusions.

Furthermore, findings based on gene expression profiling need, as any other clinical procedures do, external validation. With increasing commercial interest, however, the technology is likely to find its way into medical routine.

References

1 Alizadeh, A., M. Eisen, R. Davis, et al. 2000. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature* **403**: 503–11.

2 Alon, U., N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack and A. Levine. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA* **96**: 6745–50.

3 Ambroise, C. and G. McLachlan. 2002. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl Acad. Sci. USA* **99**: 6562–6.

4 Ashburner, M., C. Ball, J. Blake, et al. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **25**: 25–9.

5 Beissbarth, T. and T. Speed. 2004. GOstat: find statistically overrepresented Gene Ontologies within a group of genes. *Bioinformatics* **20**: 1464–5.

6 Ben-Dor, A., N. Friedman and Z. Yakhini. 2001. Class discovery in gene expression data. *Bioinformatics* **17**: 31–8.

7 Bengio, Y. and Y. Grandvalet. 2004. No unbiased estimator of the variance of *k*-fold cross-validation. *J. Machine Learning Res.* **5**: 1089–105.

8 Bhattacharjee, A., W. Richards, J. Staunton, et al. 2001. Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses. *Proc. Natl Acad. Sci. USA* **98**: 13790–5.

9 Bittner, M., P. Meltzer, Y. Chen, et al. 2000. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature* **406**: 536–40.

10 Bø, T. and I. Jonassen. 2002. New feature subset selection procedures for classification of expression profiles. *Genome Biol.* **3**: RESEARCH0017.1–11.

11 Bolshakova, N., F. Azuaje and P. Cunningham. 2005. An integrated tool for microarray data clustering and cluster validity assessment. *Bioinformatics* **21**: 451–5.

12 Braga-Neto, U. and E. Dougherty. 2004. Is cross-validation valid for small-sample microarray classification? *Bioinformatics* **20**: 374–80.

13 Breiman, L., J. Friedman, R. Olshen and C. Stone. 1984. *Classification and regression trees.* Wadsworth, Belmont, CA.

14 BREIMAN, L. 1996. Bagging predictors. *Machine Learning* **24**: 123–40.

15 BREIMAN, L. 2001. Random forests. *Machine Learning* **45**: 5–32.

16 BROWN, M., W. GRUNDY, D. LIN, N. CRISTIANINI, C. SUGNET, T. FUREY, M. ARES AND D. HAUSSLER. 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl Acad. Sci. USA* **97**: 262–7.

17 BULLINGER, L., K. DÖHNER, E. BAIR, S. FRÖHLING, R. SCHLENK, R. TIBSHIRANI, H. DÖHNER AND J. POLLACK. 2004. Use of gene-expression profiling to identify prognostic subclasses in adult acute myeloid leukemia. *N. Engl. J. Med.* **350**: 1605–16.

18 CARIO, G., M. STANULLA, B. FINE, et al. 2005. Distinct gene expression profiles determine molecular treatment response in childhood acute lymphoblastic leukemia. *Blood* **105**: 821–6.

19 CHEESEMAN, P. AND J. STUTZ. 1996. *Advances in Knowledge Discovery and Data Mining.* Bayesian Classification (AutoClass): Theory and Results. In FAYYAD, U. M., G. PIATETSKY-SHAPIRO, P. SMYTH AND R. UTHURUSAMY (eds.), AAAI Press, Menlo Park, CA: 80–153.

20 CHIARETTI, S., X. LI, R. GENTLEMAN, A. VITALE, M. VIGNETTI, F. MANDELLI, J. RITZ AND R. FOA. 2004. Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. *Blood* **103**: 2771–8.

21 CHO, J., D. LEE, J. PARK AND I. LEE. 2004. Gene selection and classification from microarray data using kernel machine. *FEBS Lett.* **571**: 93–8.

22 DAVE, S., G. WRIGHT, B. TAN, et al. 2004. Prediction of survival in follicular lymphoma based on molecular features of tumor-infiltrating immune cells. *N. Engl. J. Med.* **351**: 2159–69.

23 DENNIS, G. J., B. SHERMAN, D. HOSACK, J. YANG, W. GAO, H. LANE AND R. A. LEMPICKI. 2003. DAVID: Database for Annotation, Visualization, and Integrated Discovery. *Genome Biol.* **4**:P3.

24 DETTLING, M. AND P. BÜHLMANN. 2003. Boosting for Tumor Classification with Gene Expression Data. *Bioinformatics* **19**: 1061–9.

25 DETTLING, M. 2004. BagBoosting for tumor classification with gene expression data. *Bioinformatics* **20**: 3583–93.

26 DEVROYE, L., L. GYÖRFI AND L. LUGOSI. 1996. *A Probabilistic Theory of Pattern Recognition.* Springer, New York, NY.

27 DING, C. AND H. PENG. 2005. Minimum redundancy feature selection from microarray gene expression data. *J. Bioinform. Comput. Biol.* **3**: 185–205.

28 DONIGER, S., N. SALOMONIS, K. DAHLQUIST, K. VRANIZAN, L. SC AND B. R. CONKLIN. 2003. MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. *Genome Biol.* 4:R7.

29 DUDA, R., P. HART AND D. STORK. 2001. *Pattern Classification.* Wiley, New York, NY.

30 DUDOIT, S., J. FRIDLYAND AND T. SPEED. 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc.* **97**: 77–87.

31 DUDOIT, S. AND J. FRIDLYAND. 2002. A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biol.* 3:R36.

32 DUGAS, M., S. MERK, S. BREIT AND P. DIRSCHEDL. 2004. mdclust–exploratory microarray analysis by multidimensional clustering. *Bioinformatics* **20**: 931–6.

33 EFRON, B. AND R. TIBSHIRANI. 1997. Improvements on cross-validation: the .632+ bootstrap method. *J. Am. Stat. Ass.* **92**: 548–60.

34 EILERS, P., J. VAN HOUWELINGEN AND J. BOER. 2000. Classification of microarray data with penalized logistic regression. *Proc. SPIE* **4266**: 187–98.

35 EIN-DOR, L., I. KELA, G. GETZ, D. GIVOL AND E. DOMANY. 2005. Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics* **21**: 171–8.

36 EISEN, M., P. SPELLMAN, P. BROWN AND D. BOTSTEIN. 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci.* **95**: 14863–8.

37 Fraley, C. and A. E. Raftery. 2002. Model-based clustering, discriminant analysis, and density estimation. *J. Am. Stat. Ass.* **97**: 611–31.

38 Freije, W., F. Castro-Vargas, Z. Fang, S. Horvath, T. Cloughesy, L. Liau, P. Mischel and S. Nelson. 2004. Gene expression profiling of gliomas strongly predicts survival. *Cancer Res.* **64**: 6503–10.

39 Freund, Y. and R. E. Schapire. 1996. *Experiments with a new boosting algorithm.* In Proc. 13th Int. Conf. on Machine Learning, Bari, Italy: 148–56.

40 Friedman, J., T. Hastie and R. Tibshirani. 2000. Additive logistic regression: a statistical view of boosting. *Ann. Stat.* **28**: 337–74.

41 Fuller, G., K. Hess, C. Rhee, W. Yung, R. Sawaya, J. Bruner and W. Zhang. 2002. Molecular classification of human diffuse gliomas by multidimensional scaling analysis of gene expression profiles parallels morphology-based classification, correlates with survival, and reveals clinically-relevant novel glioma subsets. *Brain Pathol.* **12**: 108–16.

42 Furey, T., N. Cristianini, N. Duffy, D. Bednarski, M. Schummer and D. Haussler. 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **16**: 906–14.

43 Geisser, S. 1975. The predictive sample reuse method with applications. *J. Am. Stat. Ass.* **70**: 320–28.

44 Getz, G., E. Levine and E. Domany. 2000. Coupled two-way clustering analysis of gene microarray data. *Proc. Natl Acad. Sci. USA* **97**: 12079–84.

45 Golub, T., D. Slonim, P. Tamayo, et al. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**: 531–7.

46 Gunther, E. C., D. J. Stone, R. W. Gerwien, P. Bento and M. P. Heyes. 2003. Prediction of clinical drug efficacy by classification of drug-induced genomic expression profiles *in vitro. Proc. Natl Acad. Sci. USA* **100**: 9608–13.

47 Guyon, I., J. Weston, S. Barnhill and V. Vapnik. 2002. Gene selection for cancer classification using sSupport vector machines. *Machine Learning* **46**: 389–422.

48 Guyon, I. and A. Elisseeff. 2003. An introduction to variable and feature selection. *J. Machine Learning Res.* **3**: 1157–82.

49 Halkidi, M., Y. Batistakis and M. Vazirgiannis. 2001. On clustering validation techniques. *J. Intell. Inform. Systems* **17**: 107–45.

50 Hall, M. and L. Smith. 1997. Feature subset selection: a correlation based filter approach. Proc. Int. Conf. on Neural Information Processing and Intelligent Information Systems, Dunedin, New Zealand: 855–8.

51 Hartigan, J. and M. A. Wong. 1979. A *k*-means clustering algorithm. *Appl. Stat.* **28**: 100–4.

52 Hastie, T., R. Tibshirani and J. Friedman. 2001. *The Elements of Statistical Learning.* Springer, New York, NY.

53 Haven, C., V. Howell, P. Eilers, et al. 2004. Gene expression of parathyroid tumors: molecular subclassification and identification of the potential malignant phenotype. *Cancer Res.* **64**: 7405–11.

54 Hsu, C.-W. and C.-J. Lin. 2002. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Networks* **13**: 415–25.

55 Huang, X. and W. Pan. 2003. Linear regression and two-class classification with gene expression data. *Bioinformatics* **19**: 2072–8.

56 Ihaka, R. and R. Gentleman. 1996. R: A Language for Data Analysis and Graphics. *J. Comput. Graph. Stat.* **5**: 299–314.

57 Jäger, J., D. Weichenhan, B. Ivandic and R. Spang. 2005. Early diagnostic marker panel determination for microarray based clinical studies. *Stat. Appl. Genet. Mol. Biol.* **4**: article 9.

58 Jäger, J., R. Sengupta and W. Ruzzo. 2003. Improved gene selection for classification of microarrays. *Pac. Symp. Biocomput.* **8**: 53–64.

59 JOHN, G. H., R. KOHAVI AND K. PFLEGER. 1994. Irrelevant features and the subset selection problem. San Mateo, USA: 121–9.

60 KAUFMANN, L. AND P. ROUSSEEUW. 1990. *Finding Groups in Data: An Introduction to Clustering.* Wiley, New York, NY.

61 KAYNAK, B., A. VON HEYDEBRECK, S. MEBUS, et al. 2003. Genome-wide array analysis of normal and malformed human hearts. *Circulation* **107**: 2467–74.

62 KIM, Y. AND J. KIM. 2004. Gradient LASSO for feature selection. In Proc. 21st Intl Conf. on Machine learning, ACM, Banff, Canada.

63 KLUGER, Y., R. BASRI, J. T. CHANG AND M. GERSTEIN. 2003. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res.* **13**: 703–16.

64 KOHAVI, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. Proc. Int. Joint Conf. for Artificial Intelligence, Montreal, Canada: 1137–45.

65 KOHONEN, T. 1997. *Self-organizing maps.* Springer New York, NY.

66 LANGE, T., V. ROTH, M. L. BRAUN AND J. M. BUHMANN. 2004. Stability-based validation of clustering solutions. *Neural Comput.* **16**: 1299–323.

67 LEE, J., J. LEE, M. PARK AND S. SONG. 2005. An extensive comparison of recent classification tools applied to microarray data. *Comput. Stat. Data Anal.* **48**: 869–85.

68 LIU, Y. AND M. RINGÉR. 2004. Multiclass discovery in array data. *BMC Bioinformatics* **5**: 70.

69 LOSSOS, I., D. CZERWINSKI, A. ALIZADEH, M. WECHSER, R. TIBSHIRANI, D. BOTSTEIN AND R. LEVY. 2004. Prediction of survival in diffuse large-B-cell lymphoma based on the expression of six genes. *N. Engl. J. Med.* **350**: 1828–37.

70 LOTTAZ, C. AND R. SPANG. 2005. Molecular decomposition of complex clinical phenotypes using biologically structured analysis of microarray data. *Bioinformatics* **21**: 1971–8.

71 MADEIRA, S. AND A. L. OLIVEIRA. 2004. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **1**: 24–45.

72 MAN, M. Z., G. DYSON, K. JOHNSON AND B. LIAO. 2004. Evaluating methods for classifying expression data. *J. Biopharm. Stat.* **14**: 1065–84.

73 MCLACHLAN, G., K. DO AND C. AMBROISE. 2004. *Analyzing Microarray Gene Expression Data.* Wiley, New York, NY.

74 MCSHANE, L., M. RADMACHER, B. FREIDLIN, R. YU, M. LI AND R. SIMON. 2002. Methods for assessing reproducibility of clustering patterns observed in analyses of microarray data. *Bioinformatics* **18**: 1462–69.

75 MICHIELS, S., S. KOSCIELNY AND C. HILL. 2005. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet* **365**: 488–92.

76 MONTI, S., K. SAVAGE, J. KUTOK, et al. 2005. Molecular profiling of diffuse large B-cell lymphoma identifies robust subtypes including one characterized by host inflammatory response. *Blood* **105**: 1851–61.

77 MONTI, S., P. TAMAYO, J. MESIROV AND T. R. GOLUB. 2003. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning* **52**: 91–118.

78 MURALI, T. AND S. KASIF. 2003. Extracting conserved gene expression motifs from gene expression data. *pac. Symp. Biocomput.* **8**: 77–88.

79 NTZANI, E. AND J. IOANNIDIS. 2003. Predictive ability of DNA microarrays for cancer outcomes and correlates: an empirical assessment. *Lancet* **362**: 1439–44.

80 OSBORNE, M., R. PRESNELL, B. TURLACH AND A. BERWIN. 2000. On the LASSO and its dual. *J. Comput. Graph. Stat.* **9**: 319–37.

81 PARK, P. J., M. PAGANO AND M. BONETTI. 2001. A Nonparametric scoring aAlgorithm for identifying informative genes from microarray data. Proc. Pac. Symp. Biocomputing **6**: 52–63.

82 PEROU, C., T. SORLIE, M. EISEN, et al. 2000. Molecular portraits of human breast tumours. *Nature* **406**: 747–52.

83 PTITSYN, A. 2004. Class discovery analysis of the lung cancer gene expression data. *DNA Cell Biol.* **23**: 715–21.

84 R DEVELOPMENT CORE TEAM. 2005. *R: A Language and Environment for Sstatistical Computing.* Foundation for Statistical Computing, Vienna, Austria.

85 RAKOTOMAMONJY, A. 2003. Variable selection using SVM-based criteria. *J. Machine Learning Res.* **3**: 1357–70.

86 RAMASWAMY, S., K. N. ROSS, E. S. LANDER AND T. R. GOLUB. 2003. A molecular signature of metastasis in primary solid tumors. *Nat. Genet.* **33**: 49–54.

87 RAMASWAMY, S., P. TAMAYO, R. RIFKIN, et al. 2001. Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl Acad. Sci. USA* **98**: 15149–54.

88 RIPLEY, B. 1996. *Pattern Recognition and Neural Networks.* Cambridge University Press, Cambridge.

89 ROEPMAN, P., L. WESSELS, N. KETTELARIJ, et al. 2005. An expression profile for diagnosis of lymph node metastases from primary head and neck squamous cell carcinomas. *Nat. Genet.* **37**: 182–6.

90 ROTH, V. AND T. LANGE. 2004. Bayesian class discovery in microarray datasets. *IEEE Trans. Biomed. Eng.* **51**: 707–18.

91 ROTH, V. 2004. The generalized lasso. *IEEE Trans. Neural Networks* **15**: 16–28.

92 RUSCHHAUPT, M., W. HUBER, A. POUSTKA AND U. MANSMANN. 2004. A compendium to ensure computational reproducibility in high-dimensional classification tasks. *Stat. Appl. Genet. Mol. Biol.* **3**: 37.

93 SCHÖLKOPF, B. AND A. SMOLA. 2001. *Learning with Kernels.* MIT Press, Cambridge, MA.

94 SHENG, Q., Y. MOREAU AND B. D. MOOR. 2003. Biclustering microarray data by Gibbs sampling. *Bioinformatics* **19 (Suppl. 2)**: II196–205.

95 SHEVADE, S. AND S. KEERTHI. 2003. A simple and efficient algorithm for gene selection using sparse logistic regression. *Bioinformatics* **19**: 2246–53.

96 SIMON, R., M. RADMACHER, K. DOBBIN AND L. MCSHANE. 2003. Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. *J. Natl Cancer Inst.* **95**: 14–8.

97 SIMON, R. 2005. Development and validation of therapeutically relevant multi-gene biomarker classifiers. *J. Natl Cancer Inst.* **97**: 866–7.

98 SMYTH, P. 2000. Model selection for probabilistic clustering using cross-validated likelihood. *Stat. Comput.* 9:63.72.

99 SORLIE, T., R. TIBSHIRANI, J. PARKER, et al. 2003. Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proc. Natl Acad. Sci. USA* **100**: 8418–23.

100 SPANG, R., H. ZUZAN, M. WEST, J. NEVINS, C. BLANCHETTE AND J. R. MARKS. 2002. Prediction and uncertainty in the analysis of gene expression profiles. *In Silico Biol.* **2**: 369–81.

101 SPEED, T. 2003. *Statistical Analysis of Gene Expression Microarray Data.* Chapman & Hall/CRC, Boca Raton, FL.

102 STONE, M. 1974. Cross-validatory choice and assessment of statistical predictions. *J. R. Stat. Soc. B* **36**: 111–47.

103 TAN, A. C. AND D. GILBERT. 2003. Ensemble machine learning on gene expression data for cancer classification. *Appl. Bioinform.* **2**: S75–83.

104 TANG, C., L. ZHANG, I. ZHANG AND M. RAMANATHAN. 2001. Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In Proc. IEEE Int. Symp. on Bioinformatics and Bioengineering, Bethesda, USA.

105 TIBSHIRANI, R., G. WALTHER AND T. HASTIE. 2001. Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc.* **63**: 411–23.

106 TIBSHIRANI, R., T. HASTIE, B. NARASIMHAN AND G. CHU. 2002. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natl Acad. Sci. USA* **99**: 6567–72.

107 TIBSHIRANI, R., T. HASTIE, B. NARASIMHAN AND G. CHU. 2003.

Class prediction by nearest shrunken centroids, with applications to DNA microarrays. *Stat. Sci.* **18**: 104–17.

108 TIKHONOV, A. 1963. Solution of incorrectly formulated problems and the regularization method. *Soviet Math. Dokl.* **4**: 1035–38.

109 VAN DE VIJVER, M., Y. HE, L. VAN'T VEER, et al. 2002. A gene-expression signature as a predictor of survival in breast cancer. *N. Engl. J. Med.* **347**: 1999–2009.

110 VAN 'T VEER, L., H. DAI, M. VAN DE VIJVER, et al. 2002. Gene expression profiling predicts clinical outcome of breast cancer. *Nature* **415**: 530–6.

111 VAPNIK, V. 1995. *The Nature of Statistical Learning Theory.* Springer, New York, NY.

112 VAPNIK, V. 1998. *Statistical Learning Theory.* Wiley, New York, NY.

113 VON HEYDEBRECK, A., W. HUBER, A. POUSTKA AND M. VINGRON. 2001. Identifying splits with clear separation: a new class discovery method for gene expression data. *Bioinformatics* **17**: S107–14.

114 WELSH, J., P. ZARRINKAR, L. SAPINOSO, et al. 2001. Analysis of gene expression profiles in normal and neoplastic ovarian tissue samples identifies candidate molecular markers of epithelial ovarian cancer. *Proc. Natl Acad. Sci. USA* **98**: 1176–81.

115 WESSELS, L. F. A., M. J. T. REINDERS, A. A. M. HART, C. J. VEENMAN, H. DAI, Y. D. HE AND L. J. V. VEER. 2005. A protocol for building and evaluating predictors of disease state based on microarray data. *Bioinformatics* **21**: 3755–62.

116 WEST, M., C. BLANCHETTE, H. DRESSMAN, et al. 2001. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc. Natl Acad. Sci. USA* **98**: 11462–7.

117 YEOH, E., M. ROSS, S. SHURTLEFF, et al. 2002. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell* **1**: 133–43.

118 YEUNG, K., C. FRALEY, A. MURUA, A. RAFTERY AND W. RUZZO. 2001. Model-based clustering and data transformations for gene expression data. *Bioinformatics* **17**: 977–87.

119 ZHANG, H., C.-Y. YU AND B. SINGER. 2003. Cell and tumor classification using gene expression data: construction of forests. *Proc. Natl Acad. Sci. USA* **100**: 4168–72.

120 ZHANG, H., C. YU, B. SINGER AND M. XIONG. 2001. Recursive partitioning for tumor classification with gene expression microarray data. *Proc. Natl Acad. Sci. USA* **98**: 6730–5.

121 ZHU, J. AND T. HASTIE. 2004. Classification of gene microarrays by penalized logistic regression. *Biostatistics* **5**: 427–43.

27
Classification of Genes

Jörg Rahnenführer and Thomas Lengauer

1 Introduction

Microarray experiments provide simultaneous measurements of transcript levels of a large number of genes in a certain tissue and cellular condition. The potential impact of this technology on our health is twofold. It has been demonstrated repeatedly that expression patterns of samples belonging to different patients can be used for classifying these samples into distinguishable groups. This process can generate disease-specific expression profiles that potentially provide improved molecular diagnosis (see Chapter 26 for a detailed description).

The second option is to draw from a set of microarray experiments the expression patterns of single genes across different samples, and analyze and classify the resulting expression level trajectories. The goal then is to elucidate the role of specific genes in the disease process. The two main objectives of this approach are (i) to assign a function to previously unannotated genes and (ii) to group genes into functionally related groups – a concept that generalizes the notion of a molecular pathway or network. Grouping genes is much more difficult than grouping samples since, typically, the number of genes is considerably larger than the number of samples which result from different experiments. In many cases, assigning molecular function to genes just on the basis of expression data is hopeless since learning about the differences between a large number of genes based on very few experiments is an ill-conditioned problem.

In addition, prediction of gene function based on expression profiles is challenging from a biological point of view. The function of a gene is not entirely encoded in expression patterns at the mRNA level, but is also influenced by translation efficiency, post-translational modifications, and other cellular mechanisms. Therefore, sophisticated mathematical models are rarely suitable and typically not worthwhile computing.

Nevertheless, it is accepted that similar expression patterns of two genes across a sizeable number of experiments can be a clue to their functional

Bioinformatics - From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

relatedness. Increasing the diversity of the set of experiments can enhance the explanatory power of gene comparisons based on coregulation. In early studies, clustering algorithms were used to group genes based on their expression trajectories, following the approach presented in a seminal paper by Eisen and coworkers [19].

Soon the limitations of this approach became evident [32]. Also, the even more challenging goal of inferring genetic networks from expression data alone turned out to be too ambitious [26]. Instead, a more promising idea was increasingly adopted: in order to construct more complex and reliable functional models, gene expression data were supplemented with additional biological information, such as chromosomal location data [55] and shared motifs in the promoters of similarly regulated genes [42,48,67,69,74]. Coexpressed genes that share functional sites in their upstream regulatory regions are expected to be under similar regulatory regimes and thus to share a functional relationship. Subsequently, explicit models of the involved regulators have been developed [57,58] (see also Chapter 21).

Another procedure which is different, in principle, but also effective, is to use the supplementary biological data as a starting point and pursue a more hypothesis-driven approach. Given predefined sets of genes that are already known to share the same functional context, one compares a list of genes identified from a microarray study with these *a priori* groups. As functional gene sets, Gene Ontology (GO) terms [1,4,16,17], metabolic pathways [50,53, 76], and protein interaction data [29,56] have been used, for example. In this setup, single unannotated genes can be tested against the predefined functional groups, transforming the unsupervised into a supervised classification problem.

In general, the difference between unsupervised and supervised learning is that in unsupervised learning there is no *a priori* output available, whereas in supervised learning, training data are labeled with outputs that are supposed to be learned. In Section 2, we give an introduction to gene classification against the background of this important discrimination. In Section 3, unsupervised methods for grouping genes from expression data are discussed in detail. In Section 4, the supervised approach is presented, giving a detailed description of methods for predicting gene function from expression data.

2 Overview of Gene Classification Tasks

From a methodical viewpoint, classifying gene probes can be divided into two major tasks, i.e. class discovery and class prediction. Class discovery makes use of unsupervised learning methods like clustering. Class prediction is applied in a supervised setting, where certain gene groups, whose functional context is known or presumed, are given in advance and single genes have to be classified to one or several of these groups. The knowledge of functional class labels for a considerable number of genes enables the assignment of unknown cDNA sequences to one of the labeled classes. We briefly describe the ideas of both approaches.

2.1 Grouping Genes without Additional Information

To date, the function of the majority of the genes on a microarray with genome-wide measurements is not known yet. Thus, addressing the unsupervised learning problem, in which one lacks gene class labels, will continue to be a relevant task for some time to come. One major objective is to partition a set of genes into new, previously unknown functional classes on the basis of their expression patterns across a number of samples. In this context, it is important to understand and account for the difference between supervised learning and unsupervised learning. If class labels are unknown, it is difficult to ascertain the validity of inferences drawn from the obtained results – a major characteristic of unsupervised learning. Cluster analysis as the most prominent approach in this situation attempts to find regions that contain modes of the underlying data distribution. The aim is to group the genes into subsets such that those within a cluster are more closely related, in some sense derived from the expression patterns, than those belonging to different clusters. When generating the clustering, one has to deliberately make two choices, i.e. which notion of similarity to use when scoring a pair of genes and which clustering algorithm to apply for grouping genes according to this similarity measure.

The analysis of expression data has not reached maturity yet. No commonly accepted stochastic data models for gene expression measurements exist. Therefore, in general, simple interpretable methods are preferred over more complex algorithms. Unsupervised methods can be used for generating new hypotheses from expression data that afterwards can be verified with other methods from molecular biology.

2.2 Functional Predictions

The major principle of unsupervised learning is to first cluster genes into groups and then assume that genes belonging to the same cluster share some biological function, e.g. in a disease process. However, researchers are often explicitly interested in inferring the function of specific unannotated genes. In this case a more promising approach is to incorporate *a priori* knowledge about the function of already annotated genes in the analysis. A disadvantage of this supervised learning approach is that the quality of the annotations is often low, such that class labels referring to biological functions are not always reliable.

The prospects can be improved by integrating other biological data into the models and predictions. Various additional sources of information have been used. The most prominent examples are shared motifs in the promoters of genes, joint membership to a metabolic or regulatory pathway, and known interactions between the corresponding proteins.

The idea to combine different data types was further established in another approach with a different philosophy. Here, a set of genes known to be in a functional context is treated as a unit. By analyzing the expression levels of such a group of genes in concert, statistical models can be introduced. Based on this model, the statistical significance of a gene group is an indication that the function common to the genes plays some role in the underlying experiment, e.g. in the analyzed disease process. Moreover, in a second step the statistical model can be used to quantify the probability of other genes to belong to this gene group.

3 Grouping Genes on the Basis of Expression Data

3.1 Cluster Analysis

Many existing clustering algorithms have been proposed and new methods have been developed specifically for grouping genes based on expression values. For a general introduction to cluster analysis, an established statistical discipline, see Ref. [33]. A recent extensive review on clustering algorithms applied to gene expression data has been presented in Ref. [3]. For grouping genes into classes, one first has to define a measure for the similarity or distance, respectively, of two genes. Then a clustering algorithm has to be chosen. Different combinations of similarity measure and clustering scheme can lead to substantially different results. Thus the analysis has to be performed with care. In the following the basics of the most widely used methods as well as a few clustering algorithms specifically developed for the analysis of expression data are described.

3.1.1 Similarity Measures

In microarray research, the most popular measures for comparing gene expression trajectories are the Pearson correlation coefficient and the Euclidean distance. Given two genes with corresponding expression vectors x and y

over p different samples, the correlation distance is defined as:

$$\mathrm{d}(x,y) = 1 - \rho_{xy} = 1 - \frac{\sum_{i=1}^{p}(x_i - \bar{x})\,(y_i - \bar{y})}{\left(\sum_{i=1}^{p}(x_i - \bar{x})^2\right)^{1/2}\,\left(\sum_{i=1}^{p}(y_i - \bar{y})^2\right)^{1/2}}.$$

Correlation measures the linear dependence between two vectors, here the vector of expression levels for two genes. The measure assigns high similarity values if the two genes exhibit a joint trend across the samples. Thus gene pairs with different means and variances can also receive small distances – an often desired feature in the analysis of gene expression data. Sometimes it is plausible that pairs of genes with opposite trends also belong to the same biological process, e.g. if upregulation of one gene causes the simultaneous downregulation of the other gene. In this case the correlation distance can be modified to $d(x,y) = 1 - |\rho_{xy}|$. In contrast, when using the Euclidean distance:

$$\mathrm{d}(x,y) = \sqrt{\sum_{i=1}^{p}(x_i - y_i)^2},$$

two genes are only close if they have similar values for all samples. Note that after standardizing genes with respect to mean and variance across samples, correlation and Euclidean distance coincide up to a constant factor. It has been demonstrated that the choice of the similarity measure has a substantial influence on the final results [53, 54]. Various measures developed in other research fields could be reasonable alternatives, but have been rarely explored in microarray re search. The Manhattan distance:

$$\mathrm{d}(x,y) = \sum_{i=1}^{p}|x_i - y_i|,$$

for example, is the sum of absolute differences of expression values and can be used as a robust version of the Euclidean distance. This measure is less sensitive to outliers in the data.

3.1.2 Hierarchical Clustering Algorithms

The initial influential paper on analyzing results obtained from expression experiments describes the application of a hierarchical clustering algorithm for grouping genes in the budding yeast *Saccharomyces cerevisiae* [19]. In this paper, for the first time, it was demonstrated that gene clusters obtained from expression data contain genes with known similar function. Hierarchical clustering immediately became popular in the field due to two major advantages. First, the large number of genes can pose a runtime problem to more complicated clustering algorithms. Second, hierarchical clustering provides a convenient visualization – a so-called *dendrogram*.

The basic idea of hierarchical clustering is the generation of a hierarchy of nested clusterings, with the number of clusters ranging from one to the number of genes. Both agglomerative and divisive hierarchical clustering exist. In divisive clustering, iteratively best possible ways of splitting a cluster into two clusters are calculated. In the more popular agglomerative clustering, initially, each gene is assigned to its own cluster. Then, iteratively the two most similar clusters are joined until only a single cluster remains. Two clusters that are joined represent a new node of the dendrogram, and new dissimilarities between this node and the remaining clusters are calculated. In average linkage hierarchical clustering, the distance between two clusters is defined as the average distance between genes in the two clusters; in single linkage, as the smallest pairwise distance; in complete linkage, as the largest pairwise distance.

The dendrogram is an intuitive visualization of the hierarchical clustering process. It depicts the level of similarity at which two clusters are merged. A line connects clusters when they are joined. The height of this line denotes the distance between the clusters. Typically, the cluster with smaller variation, e.g. measured by within-cluster variance, is plotted on the left side, but other procedures exist (see Section 4.5 in Ref. [62]).

Figure 1 shows a dendrogram calculated from gene expression measurements for patients with acute lymphoblastic leukemia (ALL) [11]. For the 20 genes with largest variance across samples and for replicates of these genes on the array, the plot shows the result of average linkage hierarchical clustering using Euclidean distance.

It turns out that even replicates of the same gene are not always put in the same cluster, e.g. the three copies of the gene *HLA-DQB1*. The third (rightmost) replicate is joined with the other two replicates only in a late step of the (agglomerative) algorithm, although a calculation shows that its closest gene with respect to Euclidean distance actually is one of the other two copies. The reason for this discrepancy is the local iterative nature of the algorithm that can lead to low-quality clustering results on a global level.

In the list of genes arranged corresponding to the order in the dendrogram, the three replicates of *HLA-DQB1* are neighbors, which can be misleading as the three genes do not belong to the same cluster. This underlines the need for a careful interpretation and evaluation of the results. An important disadvantage of hierarchical clustering is the information loss due to the enforcement of a tree structure. However, despite all these pitfalls, due to the large number of genes, subgroups of genes with a high degree of coregulation often exist. Hierarchical clustering can be particularly useful for identifying such compact subgroups.

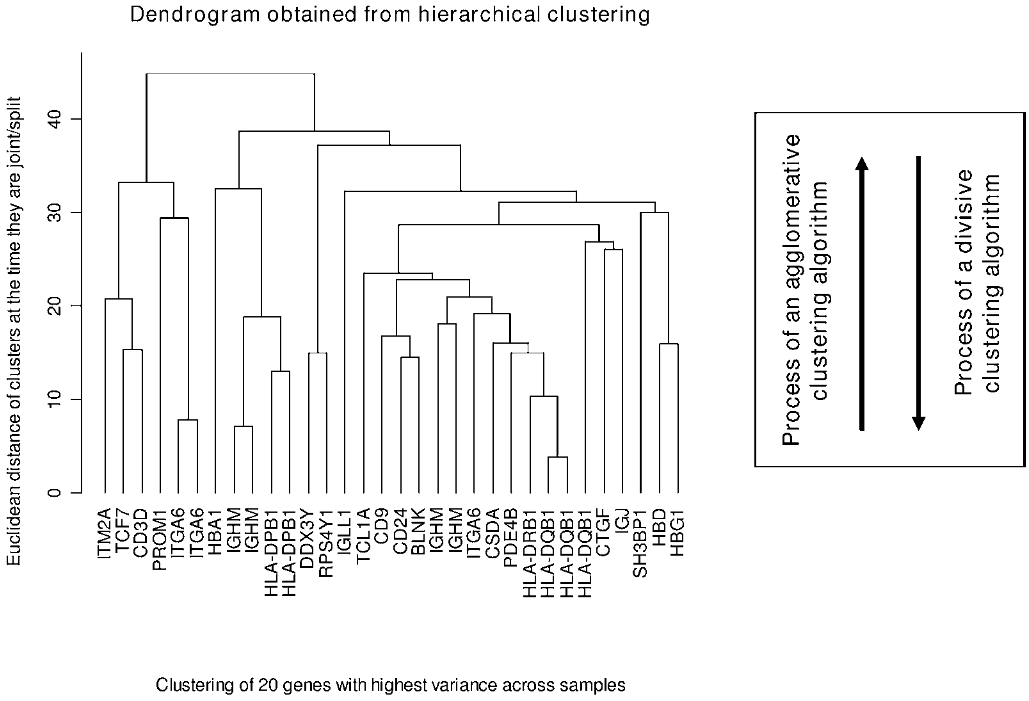


Figure 1 Average linkage hierarchical clustering of 20 high-variance genes and their replicates using Euclidean distance. Gene expression measurements are obtained from 128 patients with ALL [11].

Figure 2 shows a heatmap of the above analyzed ALL data set. A heatmap is a popular visualization for microarray data, providing a false color image of the expression values with a dendrogram added to the left side and to the top. Typically, rows correspond to genes and columns to samples.

3.1.3 **Partitioning Clustering Algorithms**

Partitioning algorithms seek to minimize the within-group dissimilarity and/or to maximize the between-group dissimilarity for a fixed number k of clusters. Often small heterogeneity within clusters is associated implicitly with a clear separation between clusters. In contrast to hierarchical clustering, the resulting clusters for different numbers of k are typically not nested.

The most popular and widely used partitioning method is k-means. This algorithm aims at minimizing the sum of within-cluster variances:

$$WSS = \sum_{i=1}^{n} \min_{j=1\ldots k} d(x_i, m_k). \tag{1}$$

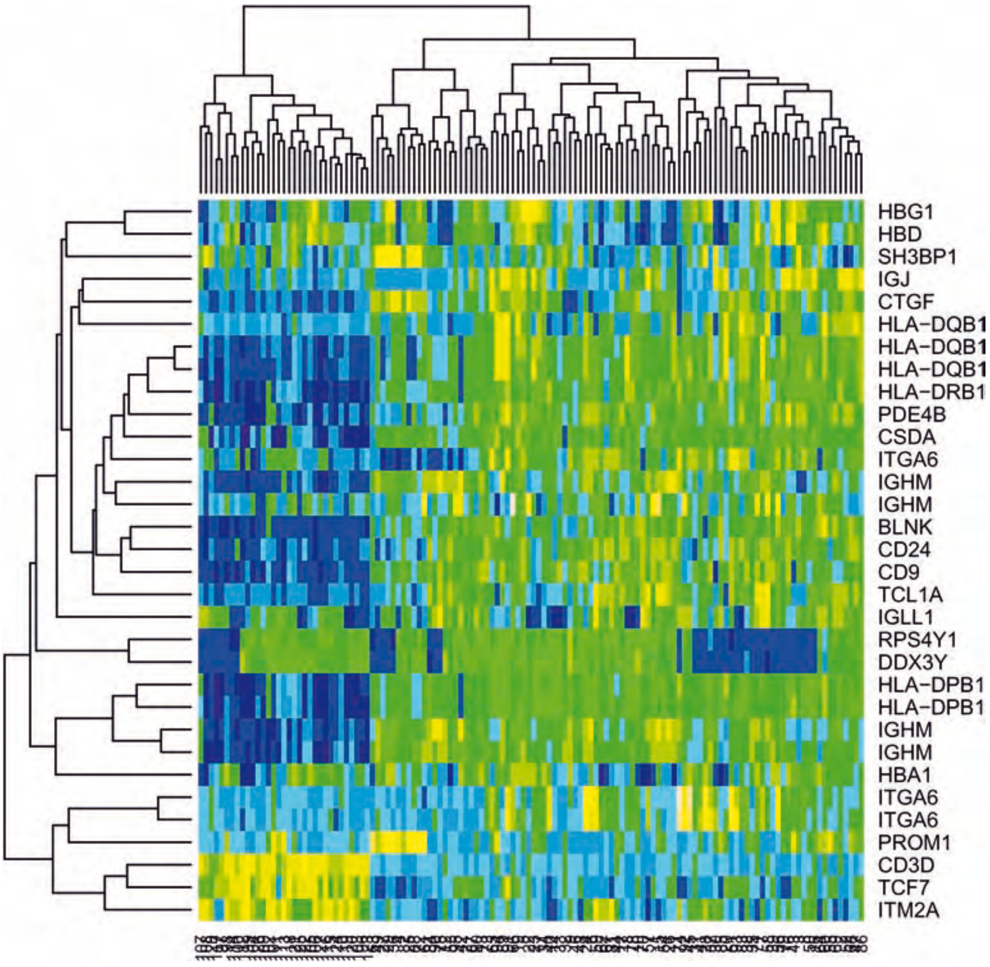


Figure 2 Heatmap for gene expression measurements from 128 patients with ALL [11].

Here, the first sum is taken over all genes, $m_1, \ldots, m_k$ are k cluster midpoints and $d(\cdot, \cdot)$ is the squared Euclidean distance. k-means starts with a random sample of k different objects as initial midpoints. Then it alternates between assigning all objects to the closest of the k midpoints with respect to Euclidean distance and calculating k new midpoints as averages of the newly assigned clusters. The midpoints are also called centroids. It is guaranteed that the algorithm converges to a local optimum. Since different starting solutions lead to different final solutions, it is advisable to run the algorithm several times and to memorize the result with minimal sum of within-cluster variances.

Several clustering algorithms can be regarded as generalizations or modifications of *k*-means. For gene expression data, especially Partitioning Around Medoids (PAM [34]) and self organizing maps (SOMs [37,64]) have been used.

PAM [34] has the advantage that it allows for an arbitrary dissimilarity matrix as input and thus is not restricted to Euclidean distance. Another important difference to *k*-means is that data points themselves serve as so-called cluster prototypes. Given pairwise dissimilarities, PAM also aims at minimizing an objective function, i.e. the sum (over all genes) of distances to the closest of *k* prototypes, see Eq. (1). In the BUILD phase, initial prototypes are chosen. Then, in the SWAP phase, iteratively the objective function is locally optimized by replacing one of the prototypes with another data point such that the objective function is decreased most. The usefulness for clustering microarray data is due to the combination of the optimization principle of *k*-means and the flexibility of hierarchical clustering algorithms regarding the choice of the gene similarity measure. A disadvantage compared to k-means can be the increased running time.

In a SOM [37] data are represented by *k* cluster prototypes that are subject to some topological restriction. The prototypes are arranged in a low-dimensional structure, typically a one- or two-dimensional array, such that every prototype has a set of neighbors. Again, a random starting solution is improved iteratively. Per iteration, one data point is picked at random and all prototypes are moved into the direction of this data point. The amount of change depends on the initial neighborhood structure of the prototypes. The prototype that is closest to the selected data point is the winner and is moved by the largest distance. The further a prototype is from the winner in the neighborhood topology, the smaller is its movement. The amount of change decreases with the number of iteration steps.

SOMs were successfully applied to group genes into biologically relevant clusters suggesting novel hypotheses about hematopoietic differentiation [64]. They are especially appealing for analyzing gene expression data since the topological structure between the prototypes provides information about relationships between neighboring clusters. On the other hand, an interpretation based on the assumption of a meaningful topology between cluster prototypes can easily be misleading. Another disadvantage of SOMs is the large number of parameters characteristic for neural networks, such that sensible parameter tuning requires some previous experience.

3.1.4 **Model-based Clustering**

Model-based clustering is based on probability models. It is assumed that the data are generated by a mixture of distributions. The most popular model is a Gaussian mixture model, in which the underlying distributions are multivariate normal distributions. The task of selecting a clustering algorithm is then reduced to fitting parameters or, more general, to a model selection problem. For the Gaussian mixture model, an Expectation-Maximization (EM) algorithm is used to estimate the means and the covariance matrices of the distributions that represent the clusters. The classical EM algorithm for learning Gaussian mixture models can be regarded as a soft version of *k*-means [24]. *k*-means assigns each sample to exactly one of the *k* clusters, whereas in the EM algorithm samples can be assigned to more than one model component. The probability of a sample to belong to a specific component is called responsibility – for a given sample all responsibilities thus add up to one. Both on real and on simulated gene expression data, model-based clustering was compared with competing clustering approaches [71]. The algorithm based on the mixture model turned out to have superior performance on selected synthetic data sets and to be competitive on real gene expression data.

The assumption that cluster analysis applied to gene expression data produces biologically related gene groups was tested in a comparative study using various clustering algorithms and gene expression data sets of different sizes [73]. The quality was measured by the fraction of gene pairs from the same clusters that share at least one known common transcription factor, making use of various transcription factor databases. On yeast data it was shown that at least 50–100 experiments are needed for identifying coregulated genes and even then gene pairs that do not share a common transcription factor are often more likely to be clustered together than coregulated genes. In this study the model-based clustering algorithm outperformed the other algorithms in terms of assigning coregulated genes to the same clusters.

In the presence of experimental replicates of genes on a microarray, the probabilistic framework of mixture models can be used for modeling between-replicates variability [44]. It was shown that for the identification of coexpressed genes improvements in precision can be achieved with as few as two replicates when the between-replicates variability is high.

3.1.5 **Biclustering Algorithms**

It is often a reasonable assumption that functionally related genes are not coexpressed in all samples of a study, but only in a subset of the analyzed samples. *Vice versa*, it is not advisable either to consider all genes when discriminating samples via expression measurements (see Chapter 26). This insight stimulated the development of so-called two-way clustering algorithms, sometimes also called *biclustering* algorithms, referring to the simultaneous clustering of genes and selection of samples. For a recent overview over biclustering algorithms see [47]. The first algorithm developed for gene expression data based on this principle was an efficient node-deletion algorithm for finding submatrices in the gene expression data with low mean-squared residue scores [10]. Another advantage of this method is that genes can belong to several clusters and thus represent multiple functions.

Progressing along this road, a graph-theoretic algorithm coupled with statistical modeling was introduced [65]. Here, both genes and samples are represented as nodes of a bipartite graph. This means that every gene is connected with every sample. Gene–sample pairs are associated with weights depending on the expression level of the respective gene in the respective sample. The algorithm determines the heaviest induced subgraph (measured in terms of the sum of edge weights) in polynomial time. It was shown that within the presented probabilistic framework it is guaranteed that the most significant biclusters are found. Moreover, using cross-validation techniques (see Chapter 26) it was demonstrated that in comparison to classical one-way clustering methods, specificity is increased when assigning gene function based on expression data.

A statistical framework for the simultaneous clustering of genes and samples was proposed [52] in which classical properties of clustering methods, such as consistency, can be analyzed. For a two-way clustering algorithm, a so-called simultaneous clustering parameter is defined. For example, this parameter can be the set of all cluster labels for genes and samples, respectively, that are obtained by a biclustering algorithm. The parameter can therefore be defined as a function of the true data generating distribution. An estimate for the parameter is obtained by applying the same function to the empirical distribution function, i.e. to the data set under investigation. Methods for estimating the distribution of the simultaneous clustering parameter are introduced. It is shown that a large number of clustering algorithms including hierarchical clustering fit this framework.

3.2 Heuristic Gene Grouping of Expression Data

The number of clustering algorithms that were developed in a variety of research areas is enormous. Still the specific structure of microarray data provoked the development of yet additional algorithms. Since the identification of gene groups with similar expression patterns in a typically small number of samples is of particular interest, a central idea is to only search for gene groups with notably low heterogeneity.

3.2.1 **CLICK Algorithm**

The CLICK algorithm (CLuster Identification via Connectivity Kernels) [59] combines probabilistic and graph-theoretic aspects. The goal is to identify groups of genes that have a high likelihood to belong to the same cluster. The input of the algorithm consists of pairwise similarities of genes. The data are represented as a graph where genes are vertices and edge weights are derived from the gene similarities. The CLICK algorithm consists of two steps. First, initial groups of genes with large pairwise similarities are identified. The algorithm recursively splits subgraphs until a stopping criterion regarding minimal gene similarity is reached or the subgraph is a singleton. In the second step, the identified gene groups are expanded to final clusters. The output comprises potential clusters with high gene similarity as well as singletons.

In a comparative study the CLICK algorithm outperformed *k*-means, hierarchical clustering and SOMs on a variety of gene expression data sets with respect to several cluster validity measures. Furthermore, it was shown that common regulatory motifs occurring in the upstream regions of coregulated genes could be identified for the resulting clusters. The suitability of the CLICK algorithm is due to its stringent criterion of within-cluster similarity, which is shown to be crucial for the relevance of inferred gene function.

3.2.2 **CAST**

CAST (Cluster Affinity Search Technique) [5] is targeted to the use on large data sets and, therefore, is also suitable for clustering gene expression data. The input of CAST consists of a gene similarity matrix and a threshold parameter. The clusters are generated iteratively. Genes are added to a cluster as long as the average similarity in the cluster exceeds the predefined threshold. If no more genes fulfill this criterion, a new cluster is opened. Afterwards, genes can still be added or re moved from clusters in order to further increase the average similarity. The algorithm is similar to hierarchical clustering in spirit, resulting in a short runtime, but the reassignment step provides additional flexibility.

3.2.3 **Gene shaving**

In a similar sense, *gene shaving* [25] aims at the identification of several small and possibly overlapping groups of genes with small between-gene variance and large between-sample variance. Principal components are orthogonal directions calculated from a data set. The first principal component is the direction in which the data have their highest variance, the second principal component indicates highest variance among all directions orthogonal to the first component and so on. In gene shaving, iteratively, clusters are generated using principal components and discarding a fraction of genes whose expression vectors have low similarity with the principal component. Both an unsupervised and a supervised version of gene shaving exist. In the supervised version, known properties of the genes are included in the process.

4 Predicting Gene Function from Expression Data

In order to produce high-quality functional predictions for genes from expression data, at least two types of information are required. First, a vocabulary of functional attributes is needed. There have been several efforts to provide meaningful directories, the most popular currently being the Gene Ontology (GO) [2] (see Chapter 29 for a detailed description). GO is a controlled vocabulary that provides structured networks of defined terms for describing attributes of gene products. Second, given such a vocabulary, one needs a considerable number of genes with annotations from that vocabulary. If both requirements are met, the supervised learning problem of assigning function to unannotated genes can be tackled by a multitude of statistical and algorithmic classification methods. If the expression pattern of a specific gene is similar to the patterns of a group of genes that are annotated with the same function, one can expect the gene to be associated with this function, as well. Another extensive source for annotations is MIPS [45] – a collection of automatically generated and manually annotated genome-specific databases and systematic classification schemes for the functional annotation of protein sequences.

The difficulties in predicting gene function from expression data have been pointed out repeatedly [35, 70]. One problem is the influence of events that happen after transcription, e.g. post-translational modifications. However, additional annotation information can help to improve the reliability of predictions. By combining the results of clustering algorithms applied to yeast genes with predefined functional annotations, useful hypotheses about protein function could be generated [70]. A variety of clustering algorithms with different parameter settings were compared regarding their ability to group coexpressed genes. Among others, hierarchical clustering, *k*-means and SOMs were applied, resulting in a large number of possibly overlapping clusters. Then, annotations and confidence values were assigned to each cluster using the hypergeometric distribution. Clusters with significant confidence values are expected to be biologically relevant. Using this procedure, potential new members of many existing functional categories involved in transcription, processing and transport of noncoding RNA molecules were found.

Even in the metabolic networks of well-studied organisms, for some reactions the corresponding enzymes have not been identified yet. In an effort to identify genes encoding such enzymes, coexpression of genes representing enzymes in close topological neighborhood within the metabolic network was used to generate predictions from expression data [35]. The method was tested by predicting metabolic enzyme-encoding genes in *S. cerevisiae* for known cases. The limitations of the approach became manifest by the negligible number of correct predictions. Only 20% of all known genes scored within the top 50 out of 5594 candidates for their respective enzymatic function. However, this number increased to 70% for those genes whose expression level had been significantly perturbed across samples. This underlines the need for considerable variation between samples.

4.1 Classification methods

For the supervised classification problem of assigning gene function in the presence of functional labels for a large training set of genes, many statistical and data-mining techniques are available. We present prototypic cases in which genes with common function could be identified.

4.1.1 Support Vector Machines (SVMs)

SVMs provide a supervised learning method that has been applied successfully to both classification and regression problems in many research areas. The basic classification SVM creates a maximum-margin hyperplane that lies in a transformed input space. Given training data points with binary labels, the maximum-margin hyperplane splits the positive and negative training points, such that the margin, i.e. the distance of the closest data point to the hyperplane, is maximized (see Chapter 26 for a more detailed introduction to SVMs).

SVMs have achieved high prediction accuracy in classifying samples based on expression data, partly due to their effective dimensionality reduction. When classifying genes, the small number of samples is typically a handicap. However, other classification methods also suffer from this dimensionality problem. The first study on gene classification that compared SVMs with other supervised learning methods, including decision trees, Parzen windows and Fisher’s linear discriminant analysis [9], was based on gene expression data from 79 samples and 2467 yeast genes with MIPS [45] annotations. Several SVMs using different similarity metrics were tested. In the study it was demonstrated that SVMs outperform the competitors with respect to correctly classifying genes to gene sets with common annotated function.

4.1.2 Rule-based Models

In order to increase interpretability, predictive rule models have been introduced for the functional classification of genes [27, 28]. The trained model defines relationships between gene expression profiles and the involvement of genes in biological processes. For example, if–then rules are used to define minimal expression profile properties needed to classify a gene to a specific biological process. The approach is supervised since functional classes of unclassified genes are learned using GO [2] annotations of classified genes.

First, biologically meaningful features are extracted from the expression data, e.g. expression increase or decrease over time in a time series experiment. Then, a rule model described in terms of these features is induced from the expression data. Before classifying unknown genes, the model is fine-turned using cross-validation on subsets of classified genes. On human fibroblast serum response expression data [32], both the predictive quality of the model and the interpretability of the extracted rules have been demonstrated.

This method provided high-precision GO biological process classifications for 211 of the 213 uncharacterized genes in the data set [38]. An advantage of the model is its capability of assigning genes to multiple biological processes. The model is flexible as it allows genes in the same functional class to exhibit a variety of expression profiles including inverse coregulation. Also, for characterized genes, new roles in biological processes were hypothesized by the algorithm and confirmed by literature search. For many previously uncharacterized genes the predicted biological processes were in agreement with homology information.

Another data-mining algorithm for predicting gene function from expression data is based on PolyFARM (Poly-machine First-order Association Rule Mining) – a program that finds first-order associations [12]. For the 40% of uncharacterized genes in the yeast *S. cerevisiae*, MIPS [45] annotations have been predicted. Again, an advantage is the informativeness of the induced rules. For many cases, agreement with biological knowledge can be observed.

4.2 Supplementing Expression Data with Additional Biological Information

In all classification methods described above the same principle is used to incorporate *a priori* biological knowledge in gene function prediction. A set of functional classes containing characterized genes is used to predict the membership of uncharacterized genes. In this sense additional biological knowledge enters only into the classification algorithm, but the unannotated genes are not augmented with other biological data. In Chapter 35, information integration for protein function prediction is described in more detail. Here, we focus on methods in which gene expression data play an important role.

One has to deliberately differentiate between two ideas for more efficiently supplementing expression data with additional biological information. In a more comprehensive classification approach, uncharacterized genes are not only associated with labeled genes, but also with other data like sequence or pathway information. The second idea is different in philosophy. The goal is to relate biological terms or functions to the underlying microarray experiments instead of characterizing single genes. For groups of genes that are associated with the same function a joint score is calculated from the expression data. If this score is significant in a statistical sense, the biological function is assumed to be relevant for the underlying experiment. In this way more subtle signals can be detected by combining small coordinated expression changes on the single-gene level to a significant change on the gene-set level.

Both ideas are different from the aim of inferring gene function by elucidating new pathways or parts thereof directly from expression data [15, 23, 29, 51, 66] (see also Chapter 21 on inferring gene regulation networks). Prominent examples are reverse engineering of genetic networks using discrete Boolean networks [15], Bayesian networks [23, 51], a model of a cellular pathway using expression data, quantitative proteomics, databases of known physical interactions [29], and a framework supporting the incorporation of biologically motivated network constraints and rules [66]. Due to the dimensionality problem with many genes and few experiments, the network reconstruction from expression data is almost infeasible [26]. In simulation experiments with dynamic Bayesian networks only local structures of the network could be recovered, but the inferred global network was meaningless. The number of spurious interactions substantially outweighed the number of true interactions.

An extension is an algorithm that identifies modules of coregulated genes, their regulators and the conditions under which regulation occurs, generating testable hypotheses [57, 58]. Also, systematic transcriptional perturbations can be used to construct better models of regulatory interactions in small networks with just few genes. The major regulatory genes in a nine-gene subnetwork of the SOS pathway in *Escherichia coli* were correctly identified with this approach [21].

A general framework for integrating external data sources with expression data is the Signature Algorithm [30, 31]. This approach can be viewed as a biclustering algorithm that is enhanced by additional biological knowledge. The standard algorithm [30] requires as input a set of genes that are expected to be coregulated in a set of experimental conditions, e.g. due to common functional annotation. First, the conditions that induce the highest average expression changes in the input group of genes are selected. Then, all genes highly expressed in these conditions are identified. The output set of genes contains a coregulated subset of the input genes, as well as correponding experimental conditions and additional genes with consistent profiles. The Iterative Signature Algorithm [31] is an extension that does not require a biologically motivated gene set. The two steps of the basic algorithm are applied iteratively until convergence. This procedure yields a self-consistent module, i.e. the resulting genes are most coherently coexpressed over the resulting conditions and the resulting conditions induce the most coherent expression of the resulting genes.

4.2.1 **Adding Sequence Data**

The most important and well-studied source for understanding gene function is sequence data. Combining chromosomal location and gene expression data, binding of gene-specific transcription activators in yeast was monitored [55]. Here, genes were identified whose expression is directly controlled by the transcription factors Gal4 and Ste12, as cells respond to changes in carbon source and mating pheromone. The identified pathways were jointly regulated by each of the two activators, and previously unknown functions for Gal4 and Ste12 were revealed. Using SVMs, gene expression profiles and phylogenetic whole-genome sequence comparisons were combined in an algorithm for function prediction [50] (see Chapter 35 for details).

Another relevant source is provided by the promoters of the investigated genes. Many studies have been dedicated to the goal of finding putative transcription factor binding sites in the upstream sequences of similarly expressed genes [48]. Correlations between known binding site motifs in the upstream regions of all genes in *S. cerevisiae* and gene expression changes in various gene-disruption experiments were uncovered. Several of these correlations turned out to be consistent with existing biological knowledge. Even without significant sequence similarity of the involved genes, on the basis of their common promoter structures gene expression data are capable of elucidating functional features of genes [69].

An ambitious task is to identify transcriptional modules, sets of genes that are coregulated in a set of experiments, through a common motif profile [58]. The motif profile specifies the relevance of different sequence motifs to the module. After an initial gene-clustering step, a sophisticated application of an EM algorithm was used to iteratively refine both the assignment of genes to the modules and the motif profile itself, in order to best explain the expression data as a function of transcriptional motifs. An evaluation of this method on two *S. cerevisiae* expression data sets demonstrated the ability to recover known motifs and to generate biologically coherent modules.

4.2.2 **Adding Gene Ontology Data**

GO [2] provides a vocabulary of gene product attributes, together with annotations of a large number of genes to these attributes. For every annotation of a gene product with a GO term, the source of this annotation and an evidence code are provided. The source may be a literature reference, another database or a computational analysis. The evidence is a categorical variable that indicates what kind of evidence is found in the cited source to support the corresponding annotation. However, for a specific gene and attribute, the crucial information is whether the gene is associated with this attribute. This information is binary and no quantitative values can be obtained. Thus, the kind of data provided by GO annotations is particularly suitable for the hypotheses-based approach in which functional gene sets are analyzed as a unity.

The approach applied most frequently in this field uses enrichment methods. In this approach the enrichment of members of a functional gene set among the top-ranking genes in a gene expression study is evaluated. Onto-Express [17] constructs functional profiles based on GO terms for the following categories: biochemical function, biological process, cellular role, cellular component, molecular function, and chromosome location. The starting point is a list of differentially expressed genes in a cellular condition under study. For a given set of functionally related genes, the significance of the number of genes contained in this list is calculated using the hypergeometric distribution or χ^2-type tests. Similar implementations of the same idea include FatiGO [1], GOstat [4], GoMiner [75] and a set of Perl modules called GO-TermFinder [6]; for an overview of current tools, see Ref. [36]. Most of these methods use Fisher's exact test for the significance calculation. We briefly describe the underlying probabilistic framework.

The statistical principle common to all scoring methods is based on the analysis of a 2×2 contingency table. Let $sigGenes$ be the set of genes identified in a microarray experiment, e.g. differentially expressed genes, and let $\overline{sigGenes}$ be all other genes on the microarray. Let $funcGenes$ be a set of genes G with a known common function and $\overline{funcGenes}$ the complement with respect to the set of all genes, denoted by $allGenes$. Table 1 counts numbers of genes according to their membership to $sigGenes$ and to $funcGenes$.

Table 1 Contingency table of genes grouped according to significance in an expression study and to membership to a functional group G.

	Significant genes	Not significant genes	Sum
Genes in G	$\lvert sigGenes \cap funcGenes\rvert$	$\lvert\overline{sigGenes} \cap funcGenes\rvert$	$\lvert funcGenes\rvert$
Genes in $\overline{G}$	$\lvert sigGenes \cap \overline{funcGenes}\rvert$	$\lvert\overline{sigGenes} \cap \overline{funcGenes}\rvert$	$\lvert\overline{funcGenes}\rvert$
Sum	$\lvert sigGenes\rvert$	$\lvert\overline{sigGenes}\rvert$	$\lvert allGenes\rvert$

Assume that a particular gene is known to have the function attributed to the gene group G. Such a gene is a member of $funcGenes$. The question then is if this gene has an increased likelihood of being a significant gene, e.g. with respect to differential expression. The significance of the dependence between the groups $funcGenes$ and $sigGenes$ can be quantified using Fisher's exact test. This test computes a p-value based on the contingency table shown in Table 1. The p-value denotes the probability of obtaining by chance at least the same amount of enrichment with significant genes as observed in the data. Lower p-values thus indicate stronger dependencies. Other tests based on the contingency table have been proposed [16]. After calculating the percentages of significant genes in *allGenes* and in *G*, respectively, *z*-scores are used to determine the significance of enrichment. From a statistical viewpoint, all these tests implicitly use random gene label assignments as the null hypothesis.

An extension of this idea is the iterative Group Analysis (iGA) that also scores functional groups like GO classes [8]. First, an ordered list of genes is obtained from the expression data. In contrast to the methods described above, no fixed cutoff in this list is predetermined for dividing the genes into interesting and uninteresting ones. Instead the algorithm moves along the list, taking into account all possible cutoffs, and calculates a *p*-value of enrichment for every cutoff. Then, the cutoff that yields the minimal *p*-value is determined and assigned to the GO group. Due to the adaptive selection of the cutoff, the minimal *p*-value itself is not valid in a statistical sense. Thus, the significance of enrichment is determined by comparing the observed minimal *p*-value with values obtained by applying the same algorithm to random gene groups *G*. The method has been extended to a graph-based group analysis [7]. For functional evidence represented in the form of graphs, e.g. metabolic or signaling pathways, subgraphs showing the most significant patterns of gene expression are identified.

The same cutoff optimization principle has been used to test a gene set for association with disease phenotype [46]. All genes are ranked according to gene expression differences in the phenotypes. The optimal cutoff in this list with respect to gene enrichment of a functional group *G* is obtained by maximizing a Kolmogorov–Smirnov running sum statistic. The significance is determined by permuting the labels for the diagnostic assignments.

GO provides a graph structure containing parent–child relationships between functional classes. If a specific functional attribute is a generalization of another attribute, then the genes in the more specific GO class form a subset of the genes in the other GO class. Efforts have been made to also integrate the topology of the GO graph into the significance calculation [41]. A graph-theoretic algorithm extracts common biological attributes of a set of interesting genes *G* in order to find the representative biological meanings. The biological significance of the group *G* then is assessed by defining a distance function on the GO graph. If the biological attributes associated with the genes in *G* are closer in GO distance, the biological interpretation is more reliable.

4.2.3 Integrating Pathway Information

Metabolic pathways are sequences of metabolic reactions that are represented by enzymes. In an initial attempt to score genes with respect to metabolic pathways based on expression data, the glycolysis pathway was analyzed

[76]. It turned out that it is not possible, in general, to assign single genes to this pathway using only gene expression measurements. However, scoring the set of genes associated to the pathway as a whole, in order to relate the pathway to the underlying microarray experiment, turned out to be a promising alternative. Using time series data, the average correlation between pairs of genes associated to the glycolysis pathway was shown to be significantly large. This approach was extended by developing scoring functions based both on coexpression and on estimated functional distance [22, 61].

In a general context, three different scores for the significance of gene groups defined by existing annotation schemes have been introduced [49]. The first score is based on the average correlation of the expression profiles in the gene group, as in previous work [76]. The second score exploits the learnability of a given classification of samples, i.e. the ability to discriminate given classes of samples using only members of the gene group. The third score compares single-gene t-test type of scores between gene class members and randomly selected genes. The latter method thus can be interpreted as an enrichment method. All methods were particularly applied to score metabolic pathways (see Chapter 35 for more details).

A sound statistical framework for calculating the significance of coregulation of genes has been presented in Ref. [53]. The ScorePAGE algorithm scores changes in pathway activity from gene expression data with a nonparametric permutation test. The algorithm was validated on two yeast gene expression data sets with time series measurements. It was shown that the specific measure for calculating coregulation, e.g. correlation or covariance, drastically influences the significance value. However, it is possible to adaptively identify a suitable measure. In addition, two extensions targeted to metabolic pathways were presented. To overcome the ambiguity of enzyme-to-gene mappings for a fixed pathway, different algorithms to select the best-fitting gene for a specific enzyme in a specific condition were introduced. It was shown that these algorithms improve the coherence of gene clusters predefined by metabolic pathways. Including information about pathway topology in the significance score further improved the sensitivity of the method. A comparison with the classical enrichment approach showed that ScorePAGE detects more relevant pathways.

4.2.4 Combination of Multiple Data Types

When combining expression data with other biological data it is reasonable not to restrict oneself to one additional data source. The main challenge for methods that use multiple sources is to combine the information from different data types in a balanced manner. In the following we describe methods that include expression data in such an analysis. A more comprehensive and detailed description of this kind of data integration is presented in Chapter 35.

In an initial attempt proteins were grouped by correlated expression patterns, correlated evolution based on phylogenetic profiles and patterns of domain fusion in order to determine functional relationships in *S. cerevisiae* [43]. For more than half of the 2557 (at that time) uncharacterized yeast proteins, a function could be assigned through a link with a characterized protein. This work represented initial evidence that general biochemical functions of proteins can be inferred by associating proteins on the basis of properties other than sequence homology.

It has been discussed how diverse large-scale data sets can be integrated to generate complex probabilistic gene networks, with a focus on exploring how these gene networks can contribute to an understanding of developmental pathways [20]. A master network was described as the total set of possible pairwise interactions between proteins encoded in a genome. Subsets of the master network are, for example, physical interaction networks of different cell types. Only a small subset of the interactions of the master network is present in any cell or tissue at any time. Thus, the idea is to first analyze the subnetworks that are important at different stages in development and to deal with the master network only at a later stage.

Another example for merging several data sources is the generation of a map of the transcriptional regulatory network in *S. cerevisiae* [74]. The presented map comprised 7419 interactions that connect 180 transcription factors with their target genes. Networks from transcription factor-binding experiments were correlated with findings from expression data. It turned out that the degree of coexpression between genes targeted by the same transcription factors increases with the number of transcription factors. Moreover, genes targeted by the same transcription factor often have similar cellular functions. A plausible insight was that, in general, correlation is not the best measure for coexpression between a transcription factor and its target, since the regulatory response of the target gene is typically delayed.

The most advanced regulatory network model generated from expression data identifies modules of coregulated genes, their regulators and the conditions under which regulation occurs [57]. The method fits in a probabilistic framework. With an EM algorithm genes are iteratively assigned to modules and the regulation model is updated for the sets of genes assigned to the modules. The result of the EM algorithm consists of testable hypotheses on which regulators regulate which models under which conditions. Functionally coherent modules and their correct regulators were identified on yeast data. The same idea was applied to identifying modules from expression data and protein interaction data [56]. Here, a module is a pathway with the two properties that genes are coregulated and the respective protein products interact. Again, using an EM algorithm, both coherent functional groups and entire protein complexes were discovered. For an overview of the combination of expression data with protein interaction data, see also Chapter 31.

The most comprehensive approach that combined multiple data types for predicting yeast protein functional classifications was based on SVMs [39, 40] (see also Chapter 35). For each data type separately, a similarity function between pairs of genes is defined. An advantage of the SVM framework is that single similarities can be combined to one function in a straightforward way. The optimal combination minimizing a statistical loss function can be computed efficiently. The method was applied to yeast genome-wide measurements of amino acid sequences, hydropathy profiles, expression data and known protein–protein interactions. In particular, the new method outperformed any of the SVMs trained on one of the single data types, which provides evidence that combining multiple data sources enhances gene function prediction.

5 Evaluation

The result of a method that classifies genes based on expression data is either a set of gene groups or a list of annotations of single genes to such groups. In both cases the results must be evaluated carefully due to inherent biological noise in expression measurements. Every algorithmic or statistical estimation process that does not account for the dependence of the estimating procedure on the data introduces too much optimism in the estimated model [24]. This effect is also called *overfitting*. Overfitting is a statistical property of estimation methods that is independent of the underlying biological truth.

In general, two kinds of evaluation procedures are available. First, other biological data or further biological experiments can be used to verify the results. However, the use of *a priori* biological knowledge for the evaluation also has disadvantages. In particular, the additional information could help in the estimation process itself (see Section 4.2 about supplementing expression data with other biological knowledge). On the other hand, the alternative approach to carry out additional experiments after the estimation process can be costly and time consuming.

The second option for the evaluation is to account for the optimism in the estimation with the help of statistical methods (see Chapter 26 for details). The most frequently applied method for eliminating optimism is cross-validation. In cross-validation, a data set is randomly partitioned into several subsets. Each of the subsets is used in turn as a test set, while the remaining subsets are aggregated and used for the estimation process. On the test set, the quality of the method can then be evaluated. An alternative is to correct for optimism by applying model selection criteria. Here, a penalty for model complexity punishes models with large parameter spaces. Due to their flexibility, more complex models would otherwise suffer from overfitting.

5.1 Assessing the Biological Relevance of Gene Groups

The biological relevance of gene groups obtained from clustering algorithms has to be checked carefully. Any clustering procedure imposes some structure on the data and groups objects into clusters, no matter whether such a structure is available in the data or not. Even if the clustering algorithm maximizes an objective function, the significance of the resulting value is *a priori* not clear. In general, one can analyze the relevance of gene groups by comparing the results with those obtained in a controlled scenario, e.g. by applying the method to data sets generated from models without the imposed structure.

5.1.1 **Validation of Clustering Results**

For validating the results obtained from clustering algorithms, one can follow three directions, using either external, internal or relative criteria. External cluster indices provide measures for the comparison with an independently drawn structure, for example based on other biological data. If such independent class labels are available, the number of misclassifications of a clustering algorithm, for example, can be computed by minimizing the number of different assignments to classes and clusters, respectively, when considering all possible matchings of class labels with cluster labels. Internal criteria assess the quality of a clustering algorithm based on the clustering itself. An example for an internal cluster index is the value of an objective function that is maximized by a clustering algorithm, like the sum of within-cluster variances (Eq. 1) for *k*-means. Relative criteria compare the clustering result with clusterings from the application of the same algorithm, but with different parameter values, or with clusterings obtained with other algorithms.

For hierarchical clustering, the coherence of the clustering result can be measured via the cophenetic distance. The *cophenetic distance* is induced by the dendrogram and assigns to a pair of genes the distance at which the respective clusters containing the two genes are merged for the first time. For validation, the cophenetic distance matrix is compared to the original distance matrix that was the basis of the hierarchical clustering. If both distance matrices are similar, the clustering captures the true structure of the data.

The importance of a careful evaluation can be discussed in the context of cell cycle genes identified by clustering expression measurements in *S. cerevisiae*. In the original publication, periodicity and correlation algorithms were applied to identify 800 genes that meet an objective minimum criterion for cell cycle regulation [63]. The gene clusters were validated by analyzing genes for known and new promoter elements that turned out to contain information predictive of cell cycle regulation. Subsequently, explicit criteria for synchronization and precise criteria for identifying gene expression patterns during the cell cycle were proposed [13]. Problems both in synchronization and statistical methodology were pointed out, causing doubt about the validity of the previous results. For identifying the correct set of periodically expressed genes, a thorough benchmark of identification methods has been provided [14], revealing that most new advanced methods perform worse than the original approach. A simple permutation-based method is proposed that performs better than most existing methods.

An initial validation algorithm for assessing the quality of clustering algorithms applied to expression data was based on cross-validation [72]. In a leave-one-out scenario the clustering algorithm is applied to the expression data with one sample left out. The predictive power of the resulting clusters is then assessed on the left-out sample. It was also shown that the quantitative measures of cluster quality obtained with this approach are positively correlated with external measures for cluster quality.

5.1.2 Estimating the Number of Clusters in a Data Set

A difficult and unsolved task is the estimation of a suitable number of clusters that is needed as input for many clustering algorithms. Two resampling methods have been proposed for estimating this number. The prediction-based method Clest [18] proves to be competitive to other estimators with respect to both accuracy and robustness. Cluster stability scores for microarray data based on a subsampling technique have been developed in the context of cancer studies [60]. Here, scores both for known and unknown clusters have been introduced. The *gap statistic* [68] compares the change in within-cluster variation with the variation that is expected under an appropriate reference null distribution. Limited applicability of the method on high-dimensional data has been discussed.

The silhouette value of an observation [34] compares the average distance of an object to members of the same cluster with the average distance to members of the closest of the other clusters. The number of clusters with minimal average silhouette width over all observations best captures the structure in the data and thus can be used as an estimate for the true number of clusters. This method can be applied in the context of any clustering algorithm. Furthermore, the silhouette value of a single observation can be used to evaluate this observation on its own. Objects with lower silhouette values are more likely to be misclassified, as they lie closer to a boundary separating two clusters, whereas a silhouette value close to 1 indicates that an object lies well in the center of the respective cluster.

5.2 Assessing Function Prediction Accuracy

The evaluation of supervised gene function prediction methods has not drawn much attention in the literature. For the classification of samples based on high-dimensional expression measurements, the need of evaluation schemes is evident. The problem of classifying samples, often to well-defined disease subtypes, is much more prevalent. Classical approaches to assess prediction accuracy, e.g. cross-validation, are described in detail in Chapter 26. These techniques have been further refined and adapted to efficiently deal with the dimensionality problem, since a meaningful gene selection is required to cope with the large number of genes compared to the small number of samples.

The classical approaches for assessing prediction accuracy can also be applied for the supervised classification of genes. However, for this task, the opposite problem emerges. The number of samples is too small compared to the number of genes. In addition, the annotation schemes are often not reliable. Thus, more importance should be attached to the biological annotations and to the number of samples. For a reliable classification, a sizeable number of samples with considerable variability in the conditions under which the expression experiments are performed is required.

6 Conclusions

Microarrays have become standard tools for gene expression profiling and their potential impact on clinical research is beyond controversy. However, different research tasks addressed with microarray technology are associated with different levels of expected clinical relevance. For the classification of patients based on expression profiles, sophisticated and suitable methods exist (see Chapter 26). In this area, the most important requirements for meaningful conclusions are studies with considerably large numbers of patients as well as careful experimental design.

Classification of genes using microarray gene expression profiles is a more difficult task due to the mostly small number of patients opposed to the huge number of genes on a standard microarray. The unsupervised task of finding coregulated gene groups depends on the choice of a distance measure between expression profiles and a clustering algorithm for grouping genes into coherent clusters. These choices can be guided by knowledge on properties of the respective measures and algorithms, but arbitrariness always remains. The supervised task of predicting gene function from gene expression data is even more difficult, mostly due to the dimensionality problem and the low quality of annotation schemes. Therefore, integrative approaches that combine expression data with other types of biological information are expected to have higher potential for reliable function prediction.

References

1 AL-SHAHROUR, F., R. DÍAZ-URIARTE AND J. DOPAZO. 2004. FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. Bioinformatics **20**: 578–80.

2 ASHBURNER, M., C. BALL, J. BLAKE, et al. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. **25**: 25–9.

3 AZUAJE, F. 2003. Clustering-based approaches to discovering and visualising microarray data patterns. Brief Bioinform. **4**: 31–42.

4 BEISSBARTH, T. AND T. P. SPEED. 2004. GOstat: find statistically overrepresented Gene Ontologies within a group of genes. Bioinformatics **20**: 1464–5.

5 BEN-DOR, A., R. SHAMIR AND Z. YAKHINI. 1999. Clustering gene expression patterns. J. Comput. Biol. **6**: 281–97.

6 BOYLE, E. I., S. WENG, J. GOLLUB, H. JIN, D. BOTSTEIN, J. M. CHERRY AND G. SHERLOCK. 2004. GO::TermFinder – open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. Bioinformatics **20**: 3710–5.

7 BREITLING, R., A. AMTMANN AND P. HERZYK. 2004. Graph-based iterative Group Analysis enhances microarray interpretation. BMC Bioinformatics **5**: 100.

8 BREITLING, R., A. AMTMANN AND P. HERZYK. 2004. Iterative Group Analysis (iGA): a simple tool to enhance sensitivity and facilitate interpretation of microarray experiments. BMC Bioinformatics **5**: 34.

9 BROWN, M., W. GRUNDY, D. LIN, N. CRISTIANINI, C. SUGNET, T. FUREY, M. ARES AND D. HAUSSLER. 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc. Natl Acad. Sci. USA **97**: 262–7.

10 CHENG, Y. AND G. CHURCH. 2000. Biclustering of expression data. Proc. ISMB **8**: 93–103.

11 CHIARETTI, S., X. LI, R. GENTLEMAN, A. VITALE, M. VIGNETTI, F. MANDELLI, J. RITZ AND R. FOA. 2004. Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. Blood **103**: 2771–8.

12 CLARE, A. AND R. KING. 2003. Predicting gene function in *Saccharomyces cerevisiae.* Bioinformatics **19 (Suppl. 2)**: II42–9.

13 COOPER, S. AND K. SHEDDEN. 2003. Microarray analysis of gene expression during the cell cycle. Cell Chromosome **2**: 1.

14 DE LICHTENBERG, U., L. J. JENSEN, A. FAUSBØLL, T. S. JENSEN, P. BORK AND S. BRUNAK. 2005. Comparison of computational methods for the identification of cell cycle-regulated genes. Bioinformatics **21**: 1164–71.

15 D'HAESELEER, P., S. LIANG AND R. SOMOGYI. 2000. Genetic network inference: from co-expression clustering to reverse engineering. Bioinformatics **16**: 707–26.

16 DONIGER, S. W., N. SALOMONIS, K. D. DAHLQUIST, K. VRANIZAN, S. C. LAWLOR AND B. R. CONKLIN. 2003. MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. Genome Biol. **4**: R7.

17 DRAGHICI, S., P. KHATRI, R. P. MARTINS, G. C. OSTERMEIER AND S. A. KRAWETZ. 2003. Global functional profiling of gene expression. Genomics **81**: 98–104.

18 DUDOIT, S. AND J. FRIDLYAND. 2002. A prediction-based resampling method for estimating the number of clusters in a dataset. Genome Biol. **3**: RESEARCH0036.

19 EISEN, M., P. SPELLMAN, P. BROWN AND D. BOTSTEIN. 1998. Cluster analysis and display of genome-wide expression patterns. Proc. Natl Acad. Sci. USA **95**: 14863–8.

20 FRASER, A. G. AND E. M. MARCOTTE. 2004. Development through the eyes of functional genomics. Curr. Opin. Genet. Dev. **14**: 336–42.

21 GARDNER, T. S., D. DI BERNARDO, D. LORENZ AND J. J. COLLINS. 2003. Inferring genetic networks and identifying compound mode of action via expression profiling. Science **301**: 102–5.

22 HANISCH, D., A. ZIEN, R. ZIMMER AND T. LENGAUER. 2002. Co-clustering of biological networks and gene expression data. Bioinformatics **18 (Suppl. 1)**: S145–54.

23 HARTEMINK, A. J., D. K. GIFFORD, T. S. JAAKKOLA AND R. A. YOUNG. 2002. Combining location and expression data for principled discovery of genetic regulatory network models. Pac. Symp. Biocomput. **7**: 437–49.

24 HASTIE, T., R. TIBSHIRANI AND J. H. FRIEDMAN. 2001. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* Springer-Verlag, New York, NY.

25 HASTIE, T., R. TIBSHIRANI, M. EISEN, et al. 2000. "Gene shaving" as a method for identifying distinct sets of genes with similar expression patterns. Genome Biol. **1**: RESEARCH0003.

26 HUSMEIER, D. 2003. Sensitivity and specificity of inferring genetic regulatory interactions from microarray experiments with dynamic Bayesian networks. Bioinformatics **19**: 2271–82.

27 HVIDSTEN, T., J. KOMOROWSKI, A. SANDVIK AND A. LAEGREID. 2001. Predicting gene function from gene expressions and ontologies. Pac. Symp. Biocomput. **6**: 299–310.

28 HVIDSTEN, T. R., A. LAEGREID AND J. KOMOROWSKI. 2003. Learning rule-based models of biological process from gene expression time profiles using gene ontology. Bioinformatics **19**: 1116–23.

29 IDEKER, T., V. THORSSON, J. RANISH, R. CHRISTMAS, et al. 2001. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. Science **292**: 929–34.

30 IHMELS, J., G. FRIEDLANDER, S. BERGMANN, O. SARIG, Y. ZIV AND N. BARKAI. 2002. Revealing modular organization in the yeast transcriptional network. Nat. Genet. **31**: 370–7.

31 IHMELS, J., S. BERGMANN AND N. BARKAI. 2004. Defining transcription modules using large-scale gene expression data. Bioinformatics **20**: 1993–2003.

32 IYER, V., M. EISEN, D. ROSS, et al. The transcriptional program in the response of human fibroblasts to serum. Science **283**: 83–7.

33 JAIN, A. K. AND R. C. DUBES. 1988. *Algorithms for Clustering Data.* Prentice-Hall, New York, NY.

34 KAUFMAN, L. AND P. J. ROUSSEEUW. 1990. *Finding Groups in Data: An Introduction to Cluster Analysis.* Wiley, New York, NY.

35 KHARCHENKO, P., D. VITKUP AND G. M. CHURCH. 2004. Filling gaps in a metabolic network using expression information. Bioinformatics **20 (Suppl. 1)**: I178–85.

36 KHATRI, P. AND S. DRAGHICI. 2005. Ontological analysis of gene expression data: current tools, limitations, and open problems. Bioinformatics **21**: 3587–95.

37 KOHONEN, T. 1990. The Self-Organizing Map. Proc. IEEE **78**: 1464—1480.

38 LAEGREID, A., T. R. HVIDSTEN, H. MIDELFART, J. KOMOROWSKI AND A. K. SANDVIK. 2003. Predicting gene ontology biological process from temporal gene expression patterns. Genome Res. **13**: 965–79.

39 LANCKRIET, G. R. G., M. DENG, N. CRISTIANINI, M. JORDAN AND W. NOBLE. 2004. Kernel-based data fusion and its application to protein function prediction in yeast. Pac. Symp. Biocomput. **9**: 300–11.

40 LANCKRIET, G. R. G., T. D. BIE, N. CRISTIANINI, M. I. JORDAN AND W. S. NOBLE. 2004. A statistical framework for genomic data fusion. Bioinformatics **20**: 2626–35.

41 LEE, S. G., J. U. HUR AND Y. S. KIM. 2004. A graph-theoretic modeling on GO space for biological interpretation of gene clusters. Bioinformatics **20**: 381–8.

42 LYONS, T., A. GASCH, L. GAITHER, D. BOTSTEIN, P. BROWN AND D. EIDE. 2000. Genome-wide characterization of the Zap1p zinc-responsive regulon in yeast. Proc. Natl Acad. Sci. USA **97**: 7957–62.

43 MARCOTTE, E., M. PELLEGRINI, M. THOMPSON, T. YEATES AND D. EISENBERG. 1999. A combined algorithm for genome-wide prediction of protein function. Nature **402**: 83–6.

44 MEDVEDOVIC, M., K. YEUNG AND R. BUMGARNER. 2004. Bayesian mixture model based clustering of replicated microarray data. Bioinformatics **20**: 1222–32.

45 MEWES, H., C. AMID, R. ARNOLD, et al. 2004. MIPS: analysis and annotation of proteins from whole genomes. Nucleic Acids Res. **32**: D41–4.

46 MOOTHA, V. K., C. M. LINDGREN, K.-F. ERIKSSON, et al. 2003. PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. Nat. Genet. **34**: 267–73.

47 OLIVEIRA, A. L. AND S. C. MADEIRA. 2004. Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans. Comput. Biol. Bioinform. **1**: 24–45.

48 PALIN, K., E. UKKONEN, A. BRAZMA AND J. VILO. 2002. Correlating gene promoters and expression in gene disruption experiments. Bioinformatics **18 (Suppl. 2)**: S172–80.

49 PAVLIDIS, P., D. P. LEWIS AND W. S. NOBLE. 2002. Exploring gene expression data with class scores. Pac. Symp. Biocomput. **7**: 474–85.

50 PAVLIDIS, P., J. WESTON, J. CAI AND W. S. NOBLE. 2002. Learning gene functional classifications from multiple data types. J. Comput. Biol. **9**: 401–11.

51 PE'ER, D., A. REGEV, G. ELIDAN AND N. FRIEDMAN. 2001. Inferring subnetworks from perturbed expression profiles. Bioinformatics **17 (Suppl. 1)**: S215–24.

52 POLLARD, K. S. AND M. J. VAN DER LAAN. 2002. Statistical inference for simultaneous clustering of gene expression data. Math. Biosci. **176**: 99–121.

53 RAHNENFÜHRER, J., F. S. DOMINGUES, J. MAYDT AND T. LENGAUER. 2004. Calculating the statistical significance of changes in pathway activity from gene expression data. Stat. Appl. Genet. Mol. Biol. **3**, No. 1, Article 16.

54 RAHNENFÜHRER, J. 2003. Efficient clustering methods for tumor classification with microarrays. 670–679. In Proc. 26th Ann. Conf. of the Gesellschaft für Klassifikation, Springer, Berlin: 670–9.

55 REN, B., F. ROBERT, J. WYRICK, et al. 2000. Genome-wide location and function of DNA binding proteins. Science **290**: 2306–9.

56 SEGAL, E., H. WANG AND D. KOLLER. 2003. Discovering molecular pathways from protein interaction and gene expression data. Bioinformatics **19 (Suppl. 1)**: i264–71.

57 SEGAL, E., M. SHAPIRA, A. REGEV, D. PE'ER, D. BOTSTEIN, D. KOLLER AND N. FRIEDMAN. 2003. Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. Nat. Genet. **34**: 166–76.

58 SEGAL, E., R. YELENSKY AND D. KOLLER. 2003. Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. Bioinformatics **19 (Suppl. 1)**: i273–82.

59 SHARAN, R., A. MARON-KATZ AND R. SHAMIR. 2003. CLICK and EXPANDER: a system for clustering and visualizing gene expression data. Bioinformatics **19**: 1787–99.

60 SMOLKIN, M. AND D. GHOSH. 2003. Cluster stability scores for microarray data in cancer studies. BMC Bioinformatics **4**: 36.

61 SOHLER, F., D. HANISCH AND R. ZIMMER. 2004. New methods for joint analysis of biological networks and expression data. Bioinformatics **20**: 1517–21.

62 Speed, T. 2003. *Statistical Analysis of Gene Expression Microarray Data.* Chapman & Hall/CRC, Boca Raton, FL.

63 Spellman, P., G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein and B. Futcher. 1998. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. Mol. Biol. Cell. **9**: 3273–97.

64 Tamayo, P., D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. Lander and T. Golub. 1999. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc. Natl Acad. Sci. USA **96**: 2907–12.

65 Tanay, A., R. Sharan and R. Shamir. 2002. Discovering statistically significant biclusters in gene expression data. Bioinformatics **18 (Suppl. 1)**: S136–44.

66 Tanay, A. and R. Shamir. 2001. Computational expansion of genetic networks. Bioinformatics **17 (Suppl. 1)**: S270–8.

67 Tavazoie, S., J. Hughes, M. Campbell, R. Cho and G. Church. 1999. Systematic determination of genetic network architecture. Nat. Genet. **22**: 281–5.

68 Tibshirani, R., G. Walther and T. Hastie. 2001. Estimating the number of clusters in a data set via the gap statistic. J. R. Stat. Soc. B **63**: 411–23.

69 Werner, T. 2001. Target gene identification from expression array data by promoter analysis. Biomol. Eng. **17**: 87–94.

70 Wu, L. F., T. R. Hughes, A. P. Davierwala, M. D. Robinson, R. Stoughton and S. J. Altschuler. 2002. Large-scale prediction of Saccharomyces cerevisiae gene function using overlapping transcriptional clusters. Nat. Genet. **31**: 255–65.

71 Yeung, K., C. Fraley, A. Murua, A. Raftery and W. Ruzzo. 2001. Model-based clustering and data transformations for gene expression data. Bioinformatics **17**: 977–87.

72 Yeung, K., D. Haynor and W. Ruzzo. 2001. Validating clustering for gene expression data. Bioinformatics **17**: 309–18.

73 Yeung, K. Y., M. Medvedovic and R. E. Bumgarner. 2004. From co-expression to co-regulation: how many microarray experiments do we need? Genome Biol. **5**: R48.

74 Yu, H., N. M. Luscombe, J. Qian and M. Gerstein. 2003. Genomic analysis of gene expression relationships in transcriptional regulatory networks. Trends Genet. **19**: 422–7.

75 Zeeberg, B. R., W. Feng, G. Wang, et al. 2003. GoMiner: a resource for biological interpretation of genomic and proteomic data. Genome Biol. **4**: R28.

76 Zien, A., R. Küffner, R. Zimmer and T. Lengauer. 2000. Analysis of gene expression data with pathway scores. Proc. ISMB **8**: 407–17.

28
Proteomics: Beyond cDNA

Patricia M. Palagi, Yannick Brunner, Jean-Charles Sanchez and Ron D. Appel

1 Introduction and Principles

The term *proteome* had its first use in 1994 to picture the *prote*in complement of a gen*ome* [89]. It describes the ensemble of protein forms expressed in a biological sample at a given point in time and in a given situation. Two years later, the term *proteomics* was first used to define the study of proteomes, in a very simplified way. A broader definition advocates that proteomics is the science that deals with the global analysis of proteins, and this includes their identification, the measure of their level of expression and their partial characterization. Other definitions state that proteomics is the large-scale study of proteins; in particular, their structures, their functions and their interactions. Whatever the definition adopted, proteomics is in constant evolution, and relies on efficient protein separation techniques, mass spectrometry (MS), bioinformatics as well as gene and protein databases.

There are numerous proteomes for a single genome and proteomes are much more complex than genomes. DNA chip technology allows for the simultaneous analysis of the expression of thousands of genes at the mRNA level and can unravel biological processes. However, the correlation between the expression of mRNA and protein is low [2]. For instance, the dynamic range between transcription factors and albumin can be 10^{12} or higher. Moreover, the complete sequences of active proteins can only be partially deduced from their corresponding gene sequence. In fact, during and after the transcription and the translation processes leading to an active gene product, alterations often occur, such as alternative splicing, N-terminal truncation and post-translational modification (PTM). In addition, single genes can be expressed in more than 20 protein forms in a single tissue, e.g. at least 22 different protein forms matching α_1-antitrypsin have been described in human plasma. Proteomics involves also the description of the events generating the modifications that these proteins carry as functional entities. Besides, one single organism will have radically different protein expressions in different stages of its life cycle and in different cells, tissues and

Bioinformatics – From Genomes to Therapies Vol. 2. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

fluids. Therefore, the analysis and the understanding of the suspected 0.5–1 million proteins in human, expressed by a number of genes that is currently estimated to be around 25 000, represent a real challenge in proteomics from the technological, analytical and bioinformatics points of view. Methods involving high-resolution protein separation, parallelization of sample preparation, automation of experimental processes and database comparison, as well as powerful and specific visualization tools had to be developed and integrated.

The development of new diagnostic tests and the discovery of new drug therapies both depend on the capacity to analyze complex systems. Proteomics offers the possibility to identify disease markers, to discover drug targets or to observe the global influence of a drug in a complex mixture of proteins. This will be possible only if one can describe the identity, the occurrence and the interaction of each individual component of this mixture.

Proteomics was a real revolution to biochemistry and molecular biology, essentially due to advances in experimental technology and the combination with bioinformatics. Due to the chemical and physical complexity of proteomes, various methodological approaches have been considered so far. Nevertheless, a representative workflow of a proteome analysis can be described as the one given in Figure 1. This pathway includes most of the wet-lab (*analytical*) and dry-lab (*bioinformatics*) steps required for the complete analysis of a proteome.

The first crucial step is the sample choice. It can be a raw biological fluid, a cell extract, a fraction of a sample, etc. Above all, the choice is also strongly dependent on the separation method to be applied since it has to be compatible with the dynamic range that the separation can handle. The proteins contained in this sample have then to be separated. In proteomics,

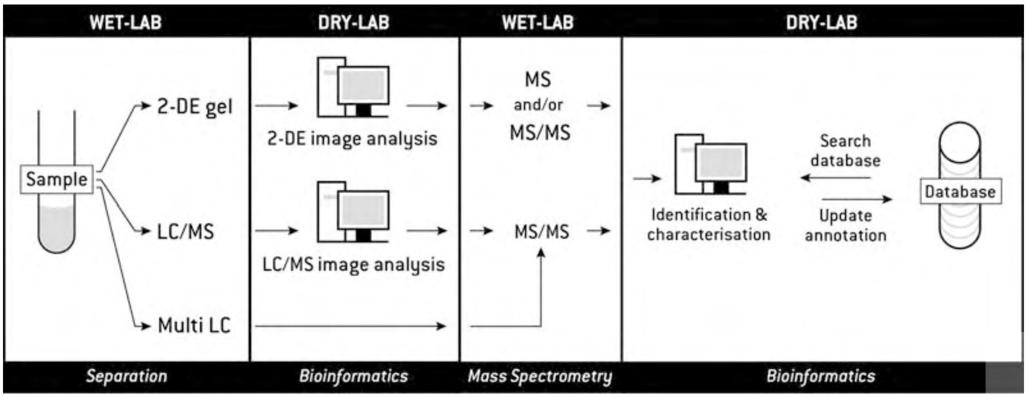


Figure 1 A schematic proteomics workflow. Digestion of proteins may occur either before the sample separation or before the MS.

one-dimensional, two-dimensional or capillary electrophoresis (respectively, 1-DE, 2-DE and CE), or liquid chromatography (LC), or a combination of them, are the preferred methods.

Once a separation method has been chosen, the next step is the analysis of the result and the selection of the proteins to be identified. In the case of a separation by 2-DE gels, the analysis is made with image analysis software. This kind of software allows for visualizing images, for comparing images and for performing a number of comparative analyses that enable tracking statistically significant changes in protein expression between populations of gels/samples. It helps to highlight the proteins of interest. Computer analysis of proteomics images is discussed in Section 3.

To proceed further, proteins separated with LC or those selected from a gel analysis are submitted to post-separation analysis. This experimental step determines highly specific protein attributes, such as peptide mass fingerprints or amino acid sequence information, when preceded by endoproteolytic cleavage. In endoproteolytic cleavage the proteins are typically incubated with an enzyme that recognizes particular amino acids and cuts the polypeptide chains at specific cleavage sites. The reaction produces shorter peptides that are fragments of the so-called digested proteins. Today's standard procedure most often involves a protein digestion step with trypsin and the analysis of the generated fragments with a crucial tool in proteomics, i.e. MS. Separation and post-separation techniques are briefly described in Section 2.

The intensive use of LC and MS in proteomics analysis has opened a new domain in proteomics imaging. The representation of LC-MS datasets as 2-D plots highlights the redundancy of data not necessarily observable when displayed in 1-D. Even though the analysis of this kind of images is still in its infancy, their potential and advantages can already be anticipated. Computer analysis of proteomics images is also discussed in Section 3.

Once in possession of the protein attributes acquired through the previous experimental steps, we can then move to database search. This search identifies a protein by looking at the best match between experimental data and data obtained by *in silico* processing and "digestion" of proteins in a sequence database. The identification (determining the name or sequence of the known proteins) and characterization (obtaining information about their function, cellular localization, PTMs, etc.) procedures using bioinformatics tools are the topic of Section 4.

Comprehensive sequence databases are a prerequisite for successful protein identification, and data from the identified proteins and samples are in turn used to populate specific proteomics databases. Section 5 browses some of the necessary databases for current proteomics projects.

The representation of a proteome analysis pathway such as the one given in Figure 1 assembles the different steps required to perform the identification of proteins from a crude biological sample. This pathway can, for instance, generate a systematic description of a complete proteome observable in a 2-DE gel. The result of such an analysis can be made concrete by the creation of an annotated database such as SWISS-2DPAGE [43] or other 2-DE databases. In addition, the information in such databases, e.g. annotated 2-DE gels, can be compared to an unannotated image to correlate positions and intensities of protein spots. From all the spots in the unannotated image, only those representing a real interest are further analyzed with identification methods. A widely used method to search for biological markers of specific diseases is the comparison of a statistically significant number of 2-DE images from samples of healthy and diseased patients or samples treated and not treated with target drugs. The images are compared, clustered and searched for protein spots that appear to be differently expressed. These become the spots of interest to be further identified. In this approach most of the efforts are concentrated in the generation of the 2-DE gels and in the comparison of the generated images. Section 3 describes the possibilities of undergoing such an analysis using dedicated software.

In addition, beyond the identification efforts, it is of great interest to describe and understand the modifications carried by the active gene products. This implies the search for splicing variants, amino acid mutations or PTMs that characterize a protein. Often, MS is used in a so-called MS/MS mode to decipher the spectra generated by this process, and to reveal structural information on the amino acid sequence and the description of PTMs attached to the studied peptides.

2 Proteomics Analytical Methods

2.1 Electrophoresis Gels

2-DE is currently one of the fundamental tools to display and evaluate the proteome complexity of any organism. It enables the separation of heterogeneous mixtures of proteins on a single polyacrylamide gel according to isoelectric point and molecular weight. The introduction of 2-DE in 1975 by O'Farrell and coworkers [48, 62, 75] allowed for separating between 2000 and 3000 proteins in a single gel. Isoelectric focusing (IEF) is the electrophoretic technique in which the proteins are separated according to their isoelectric point (pI) through a pH gradient. The pI of each protein corresponds to the pH at which the net charge of the protein is equal to zero. Under an electric field the proteins migrate until they reach their pI. Two methods are available for IEF. In the first one, the pH gradient is created and maintained by passing an electric current through a solution containing amphoretic compounds. These compounds are molecules with particular p*K*s. By mixing them in the polyacrylamide gel, it is possible to form a mixture, which establishes a pH gradient. In the second IEF technique, an immobilized pH gradient (IPG) is used [36]. In this process, the pH gradient, which exists prior to the IEF, is copolymerized within the fibers of the polyacrylamide matrix. This type of pH gradient is achieved through the use of a set of weak acids and bases named immobilines. Different types of IPG strips can be used with different ranges of pH. Most of the laboratories running 2-DE gels are currently using strips with a pH range from 3.5 to 10 to display a wide range of proteins. However, IPG strips with a narrow pH range can also be used to create a biochemical zoom on a part of the gradient range (like 4–7 IPG or 2–5 IPG). This provides higher protein concentration loading and the investigation of more proteins on a proteome "window".

After the first dimension of separation (IEF), the IPG gel containing proteins needs to be equilibrated in a solution containing sodium dodecylsulfate (SDS). It is an anionic detergent that binds to the majority of proteins in a constant mass ratio (1.4 g of SDS per gram of proteins). SDS allows the proteins to acquire a net charge per mass unit that is approximately constant. During this equilibration step, proteins are first reduced using dithiothreitol and then alkylated by incubation with iodoacetamide. Once the equilibrium is achieved, the IPG strip is loaded on the top of a polyacrylamide gel. In this second dimension of the 2-DE technique, the polypeptides enter into the polyacrylamide gel under an electric field and migrate through the porosity strictly according to their size (the molecular weight, M_r).

After the second dimension, the polypeptides must be detected either by staining, radiation (for radiolabeled proteins) or immunologic assays. Silver staining is one of the most popular methods of detection due to its very high sensitivity (below 1 ng). The main drawback of this method is the incompatibility with MS because of the presence of glutaraldehyde, which cross-links proteins. To overcome this problem, a silver staining method without glutaraldehyde has been introduced (about 10 times less sensitive) [91]. Coomassie blue staining is also very popular. It is a very simple method, although 50 times less sensitive than standard silver staining. Other methods are also used such as fluorescence or negative staining with zinc. After staining, the 2-DE gels can be scanned, and are ready to be analyzed through informatics software developed to perform quantitative protein analysis and automatic comparison of gels.

Even though 2-DE gels are considered the gold standard of proteomics, some issues remain uncertain. Due to analytical problems, between-gel variations (up to 30% coefficient of variation) lead to the problematic detection and quantification of differences in protein expression, resulting in difficulties in distinguishing biological from experimental variations. The 2-D difference gel electrophoresis (DIGE [83]) partially circumvents these problems. The DIGE method labels each sample with a spectrally resolvable fluorescent dye (Cy2, Cy3 or Cy5) prior to electrophoresis. The labeled samples are then mixed before IEF and resolved on the same 2-DE gel. Different images are acquired using different wavelengths and are compared with image analysis software. Variation in spot intensities due to gel-specific experimental factors is the same for each sample within a single DIGE gel. Consequently, the relative amount of a protein in a gel in one sample compared to another is unaffected.

Other issues remain unsolved in the 2-DE technique. This approach does not allow for identifying and characterizing all proteins present in a biological sample. Some proteins will not appear on the gel because of their extreme physicochemical properties, such as exceedingly high or low molecular weight, exceedingly high pI, extreme hydrophobicity, or insufficient abundance. Other techniques must therefore be applied to complete the picture, such as 1-DE (which separates samples only by molecular weight), pretreatment of the starting material, prefractionation and early digestion of the protein mixture followed by LC.

2.2 LC

The first chromatographic experiments were carried out in 1906 by the Russian botanist Mikhaïl Tswett, when he succeeded in separating various vegetal pigments. Chromatography is an analytical method for separating the different components of a mixture of molecules. It relies on differential affinities of these molecules for a mobile phase, the medium that carries the molecules, and for a stationary adsorbing medium, through which they pass. The stationary phase is either a solid or a liquid and the mobile phase is either a gas or a liquid. The separation is achieved through the different rates of migration of each component of a complex mixture, depending on their affinity for the stationary phase. Three types of chromatography have been developed: thin-layer (TLC), gas (GC) and liquid (LC) [27,49,52,70].

In LC the mobile phase is a liquid and the stationary phase can be either a liquid or a solid. The role of the liquid mobile phase is first to solubilize the protein sample and second to carry the proteins along the column. Various types of LC have been developed, depending on the type of stationary phase. The separation can be based on partitioning, adsorption, ion exchange, size exclusion or affinity.

In partitioning LC, the stationary phase is composed of an organic liquid covalently fixed on an inert solid support (such as silica) or embedded in a porous inert support. The principle of separation is based on the partitioning coefficient of the proteins between the nonmiscible stationary and mobile phases. The higher the partitioning coefficient, the higher the number of soluble proteins in the organic stationary phase and the higher the retention time for these proteins [10].

In adsorption LC, the stationary phase is composed of a solid adsorbent. The proteins are separated due to their varying degree of adsorption onto the solid surfaces. Different types of adsorbent can be used. In normal phase absorption, the stationary phase is composed of a polar compound (such as an amine or carboxyl) linked to silica beads. In contrast, in reverse-phase LC the stationary phase is composed of a nonpolar hydrophobic compound. In general, the hydrophobic compounds are aliphatic chains such as C_8 or C_{18} bound to silica beads. When the proteins are loaded to the top of the column they bind to aliphatic compounds through hydrophobic interactions. The proteins are eluted with increasing concentration of a hydrophobic solvent such as acetonitrile. The low-hydrophobic proteins are eluted first by low concentration of solvent and the high-hydrophobic proteins are eluted only by a high concentration of solvent [61].

Ion-exchange LC is used to separate proteins on the basis of their electric charge. In this method, ionic groups are covalently bound to the matrix column. These groups can be either positively or negatively charged and are compensated for by small concentrations of counterions that are present in the buffer. When a sample is added to the column, an exchange with the weakly bound counterions takes place. With a cationic exchange column the matrix is negatively charged. The higher the positive charges of proteins, the higher are their ionic interactions with the matrix. Increasing concentrations of salt solution are then used to elute the bound peptides. The principle is the same with anionic exchange, except that the matrix will be positively charged and the most negatively charged proteins have higher retention times [78].

Size-exclusion LC is used to separate proteins by their size and shape. Its principle is to use a porous polymer composed of beads with very small holes. As the protein solution is poured on the column, small molecules enter the pores in the beads. The small molecules are eluted last because they have a longer path to go through, as they get stuck over and over again in the maze of pores running from bead to bead [9].

Finally, affinity chromatography relies on the biological ability of a protein to bind to a column. The most common type involves a ligand, a specific small biomolecule. This small molecule is immobilized and attached to a column matrix, such as cellulose or polyacrylamide. The target protein is then passed through the column and binds to it by its ligand, while other proteins elute out. This is a very efficient purification method since it relies on the biological specificity of the target protein, such as the affinity of an enzyme for its substrate [50].

Currently, a large number of proteomics projects couple LC with MS, which is described in the following section.

2.3 MS

MS is an analytical technique that is commonly used to identify unknown compounds, quantify known materials, and elucidate the structural and physical properties of ions. It is a technique associated with very high levels of specificity and sensitivity. Analyses can often be accomplished with tiny quantities, sometimes requiring less than picogram amounts of material. Sir J. J. Thomson developed the first mass spectrometer in the first decade of the 20th Century, even before the determination of mass-to-charge ratios (m/z) of ions. The goal of all mass spectrometers is the exact determination and analysis of the m/z ratio of ions. A mass spectrometer is formed of three fundamental parts, i.e. the ionization source, the analyzer and the detector. The first step in a mass spectrometric process is to generate ions from the analyte. Nowadays, the two ionization methods used most frequently for biochemical analyses are electrospray ionization (ESI) and matrix-assisted laser desorption ionization (MALDI). However, other methods also exist such as chemical ionization (CI), electronic ionization (EI) or fast atom bombardment (FAB).

ESI, developed by Fenn and colleagues [30], is one of the atmospheric pressure ionization (API) techniques and is well suited for the analysis of polar molecules ranging from less than 100 to more than 10^6 Da in molecular weight. In ESI, the sample is introduced into a capillary and then submitted to a high voltage (around 4000 V). The high voltage causes the formation of a cloud of charged droplets. Evaporation of the solvent from the initially formed droplet as it traverses a pressure gradient toward the analyzer leads to a reduction in diameter and an increase in surface field until the Rayleigh limit is reached. The latter corresponds to the point where the surface tension can no longer sustain the charge. At this point, an explosion occurs and the droplet is ripped apart. This produces smaller droplets that can repeat the process until obtaining singly or multiply charged ions.

The Nobel Laureate, Koichi Tanaka [80], developed the MALDI method in 1987. It is based on the bombardment of molecules with a laser light to provoke sample ionization. Most commercially available MALDI mass spectrometers now have a pulsed nitrogen laser of 337 nm wavelength. In MALDI, the sample is first mixed with a highly absorbing matrix compound. The matrix cocrystallizes with the sample on the metal support. The matrix minimizes the degradation of the sample caused by the energy absorption of the incidental laser beam. The exact mechanism by which the MALDI ionization occurs is not precisely understood. It is supposed that the matrix absorbs the energy transmitted by the laser and this contribution of energy causes its expansion in gas phase by involving the molecules of the sample. The sample is mainly ionized by transfer of protons, either before desorption in the solid phase or by collision after desorption with the excited matrix to form singly charged ions.

Once obtained, ions must be analyzed. The analyzer uses dispersion or filtering to sort out ions according to their m/z ratios. Several types of analyzers have been developed such as the quadrupole, the ion trap or the time of flight (TOF). The Quadrupole mass filter consists in four parallel rods to which an electric field is applied. The resultant magnetic field allows for selecting ions according to their m/z ratio. Indeed, at one particular magnetic field strength, only ions with a particular m/z will pass trough the quadrupole filter and all other ions are thrown out of their original path. The ion trap mass analyzer consists of three hyperbolic electrodes: the ring electrode, the entrance end-cap electrode and the exit end-cap electrode. These electrodes form a cavity in which it is possible to trap and analyze ions. Both end-cap electrodes have a small hole in their center through which the ions can travel. Ions produced from the source enter the trap through the inlet focusing system and the entrance end-cap electrode. Various voltages can be applied to the electrodes to trap and eject ions according to their m/z ratios.

TOF mass analyzers separate ions by virtue of their different flight times over a known distance. A brief burst of ions is emitted from a source. These ions are accelerated so that those with the same charge have an equal kinetic energy and then are directed into a flight tube. Since kinetic energy is equal to $1/2mv^2$ (where m is the mass of the ion and v is the ion velocity), the lower the mass ion, the greater is the velocity and the shorter is the flying time. The travel time from the ion source through the flight tube to the detector, measured in microseconds, can be transformed onto an m/z value through the relationships described above. As all ion masses are measured for each ion burst, TOF mass spectrometers offer high sensitivity as well as rapid scanning. They can also provide mass data for very-high-mass biomolecules [79].

Two analyzers can be combined to perform tandem MS (MS/MS). The first analyzer selects a first ion, which is called the "parent ion". After fragmentation of the parent ion by collision-induced dissociation (CID – a fragmentation obtained by the collision of a molecule with neutral gas molecules), a second analyzer measures the m/z ratio of the ions resulting from this fragmentation. As fragmentation occurs mainly in the peptide bond of the amino acid backbone, a ladder of sequence ions is generated. The resulting peptide fragmentation masses differ by the mass of amino acid residues, thus allowing stretches of the peptide sequence to be deduced, which can be very useful to identify compounds in complex mixtures [1]. Fragments will only be detected if they carry at least one charge. According to Roepstorff's nomenclature [74],

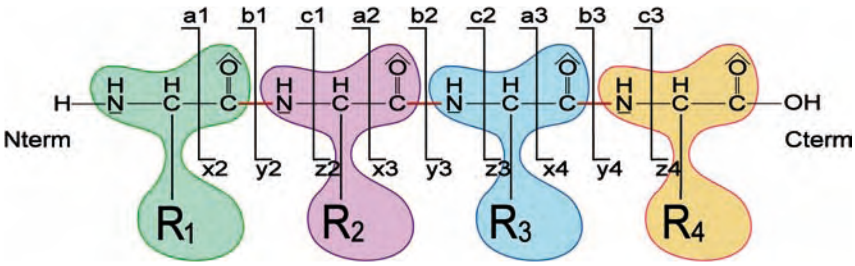


Figure 2 Fragmentation schema of a peptide with four amino acids. Fragments that carry the charge on the N-terminal side are denoted a, b and c, while those carry the charge on the C-terminal are denoted x, y and z.

the product ions are denoted as a, b and c, when the charge is retained on the N-terminal side of the fragmented peptide, and x, y and z when the charge is retained on the C-terminal side. As shown in Figure 2, ion types differ by the position of the fragmentation in respect with the peptide bond.

Once the ion passes through the mass analyzer, the ion detector (the final element of the mass spectrometer) then detects it. The detector allows a mass spectrometer to generate an electric current signal from incident ions by generating secondary electrons, which are further amplified. Alternatively, some detectors operate by inducing an electric current generated by a moving charge. Among the described detectors, the electron multiplier and scintillation counter are the most commonly used and convert the kinetic energy of incident ions into a cascade of secondary electrons.

Ideally, the MS and MS/MS processes should generate a list of m/z values of all peptides (respectively, fragmented ions) present in the analyzed sample. In reality, due to physicochemical properties of some peptides or ions, not all m/z values are detected making the data interpretation much harder.

Finally, MS can be used in combination with analytical separation methods such as LC (LC-MS) or GC (GC-MS). In these cases MS is used as a detector after chromatographic separation. Nowadays, this configuration is often used in many laboratories because it allows for combining high separation and detection capacities [22]. Shotgun proteomics has evolved from the combination of some of these techniques, MS and multidimensional chromatography, and bioinformatics. In a typical shotgun proteomics approach, a complex protein mixture is digested with proteases to produce an even more complex peptide mixture. The peptides are loaded directly onto an LC/LC column placed in-line with an MS/MS spectrometer. The spectra are acquired "on the fly" as peptides are eluted from the column, ionized, and emitted into the mass spectrometer. Using elaborated algorithms, respective peptide sequences generated from MS/MS are automatically identified by comparison against protein databases, avoiding any manual interpretation. One such approach is the multidimensional protein identification technology (MudPIT [86]), which has been applied to the proteome study of various species such as yeast, *Escherichia coli* [72], human extracts [14], *Toxoplasma gondii* [93], etc.

2.4 Protein Chips

The necessity to miniaturize and to automate high-throughput analysis systems led to the development of microarray (biochips) technology. It is based on immobilizing small molecules, oligonucleotides or proteins onto surfaces for various high-throughput screening studies. Microarrays were initially developed for large-scale gene expression studies, e.g. to compare gene expressions between different tissue types, treatments, disease models and human samples (see Chapters 24–27). Today, DNA microarrays are well established for studying the transcriptional state of a biological sample [69]. However, the level of expression of a transcribed mRNA does not always correlate with the protein expression level [39]. Protein microarrays were thus developed to analyze the expression level of a large number of proteins simultaneously and to study the interaction of proteins with a variety of molecules. It is already a successful method for the identification, quantitation and functional analysis of proteins in proteome research [55].

There are two main types of protein microarrays used: (i) to measure the abundance of a protein (abundance-based microarray) by the use of a specific reagent, and (ii) to study protein function. Abundance-based microarrays are divided in two types: capture microarrays and reverse-phase microarrays. In capture microarrays, a molecule (such as antibodies, peptide, RNA or DNA aptamers, or chemical molecules) is spotted on the surface of the microarray to catch and assay their target from a complex mixture. The relative amounts of the targeted proteins are then determined by comparison with a reference sample [40]. In this technique, the most popular capture molecules are antibodies because of their sensitivity and selectivity. However, it is estimated that only 30% of the commercially available antibodies can be used qualitatively and only 20% quantitatively. Contrary to capture microarray, reverse-phase microarrays [76] consist of spotting an unknown mixture sample on the array and probing it with an antibody or another specific reagent.

In function-based microarrays, the protein of interest is spotted on the microarray surface to study its biochemical properties and/or activities. This type of microarray can be used to examine protein interaction with other proteins, nucleic acids, lipids and other biomolecules. Furthermore, function-based microarrays can be used to study enzyme activity and substrate specificity [94, 95]. However, there are several limitations to this type of microarrays. First, proteins are often produced *in vitro* and do not possess their native conformation and their usual PTMs. Furthermore, the methods used to attach proteins to the surface array can change the behavior of the proteins. Finally, all tests are realized *in vitro* and must be confirmed *in vivo*.

Whilst there is tremendous excitement about the potential of protein arrays to further our understanding about protein function, protein–protein interactions and biological activities, there are also concerns over frustrating technical limitations. Robustness, sensitivity and automation of protein microarrays still need to be improved in order to increase its use in proteomics. The bioinformatics issues of protein microarrays are similar to those of cDNA microarrays (see Chapters 24–27) and they are deliberately not dealt with in this chapter. Several other proteomics approaches and combinations of methods exist to separate, analyze and quantify complex protein mixtures. They are not treated in this chapter either. We have intentionally decided to focus on a few proteomics techniques that touch especially challenging bioinformatics issues, which are detailed in the following sessions.

3 Computer Analysis of Proteomics Images

For each analytical method (2-DE, LC, MS and MS/MS) described in the previous session, there is a corresponding bioinformatics tool to analyze and interpret the resulting data. These tools allow for detecting differentially expressed proteins in different proteomics samples, on the one hand, and for identifying and characterizing the most pertinent proteins, on the other hand.

3.1 Analysis of 2-DE Gels

2-DE gel patterns, once digitized, provide an important basis for quantitative analysis and comparative proteomics. The possibility of detecting protein expression changes associated with diseases and treatments or finding therapeutic molecular targets opens up new frontiers for biological and biomedical research. These applications have been a major incentive to the development of specialized software systems for 2-DE gel image analysis [3, 5, 34, 51, 84]. Currently, a number of dedicated software packages are commercialized, the main ones being listed in Table 1. Although each of the 2-DE gel image analysis systems has its own philosophy and approach [85], most of them provide the same basic operations and functionalities necessary to carry out a complete gel study. These operations are usually based on state-of-the-art image processing algorithms that have been adapted to this specific biological need. In this section, the key issues and steps of 2-DE gel image analysis are discussed and illustrated using the ImageMaster^{TM} 2D Platinum release 6 (powered by Melanie, which is developed by the Proteome Informatics Group at the Swiss Institute of Bioinformatics). Note that a free viewer of this software is available at www.expasy.org/melanie. It has the usual visualization operations of the full version and most of the analysis procedures as well, even though the analysis is restricted to a small number of proteins and only from gels that have already been analyzed by a full version.

Table 1 Major commercialized 2-D image analysis software

Software	Company	Source website
DeCyder	GE Healthcare (formerly Amersham Biosciences)	www.gehealthcare.com
Delta2D	Decodon	www.decodon.com
ImageMaster^{TM} 2D Platinum	GE Healthcare	www.gehealthcare.com
PDQuest	Bio-Rad	www.bio-rad.com
Progenesis/Phoretix	Nonlinear Dynamics	www.nonlinear.com
Proteomweaver	Definiens	www.definiens.com
Z3/Z4000	Compugen	www.2dgels.com

3.1.1 Data Analysis and Validation

The first operation of a 2-DE gel analysis is the scanning of the gels. In most cases, this is achieved by the use of flatbed document scanners, camera systems, densitometers, phosphor imagers or fluorescence scanners. These equipments produce images of typically around 2000 × 2000 pixels or more and a depth of 12 or 16 bits, thus providing a dynamic range of 4096 or 65 536 gray levels, respectively.

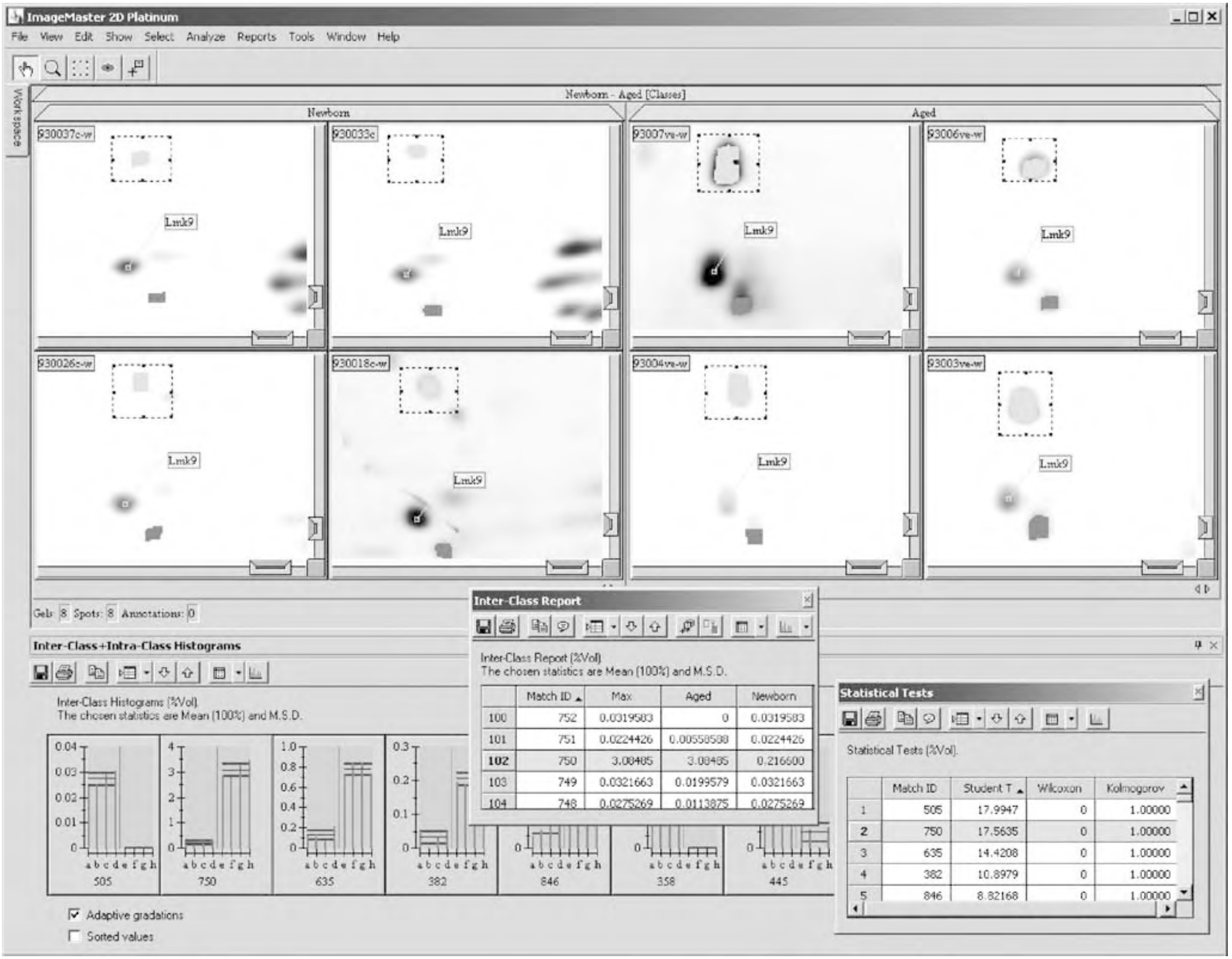
One of the basic properties of 2-DE analysis software is image visualization. ImageMaster provides several means of manipulating and displaying the gels, such as gel stacking, various zoom modes, customizable grids, "flicking" back and forth between gels or a transparency mode to visually inspect the result of matching. This makes it easy to browse through the data extracted from the 2-DE images.

Apart from the basic visualization properties, the major functions of software systems for 2-DE image analysis are (i) to detect and quantify the protein spots on the gels, (ii) to match corresponding spots across gels and (iii) to locate significant protein expression changes. To achieve (iii), both (i) and (ii) must have been successfully carried out. The optimal and reproducible definition of spot borders depends mostly on gel running issues, uneven focusing and polymerization problems. Very often proteins are not resolved as discrete spots, particularly in regions with high spot density. Numerous dim spots may be missed because they are confused with the background, whilst others might be wrongly detected. In order to overcome these issues, detection algorithms often include filtering steps to automatically remove streak artifacts and noise spikes [6] or a segmentation process based on the analysis of the gray levels [23]. Spot detection algorithms produce a repository of all protein spots contained in the 2-DE images, as well as related quantitative data, such as the spots' optical densities, area and volume (integration of the optical density over the area). Relative measures of these values are also given, e.g. the relative volume calculated as the absolute volume divided by the total volume of protein in the whole 2-DE gel. Relative values allow for partially compensating for variations in sample load or staining. Using such relative quantitative values provides better reproducibility of data.

Finding corresponding pairs of spots in gel images is also a critical task, whether it is based on the detection of spots first [68] or based on the intensities of the regions before the detection of spots [77]. Pair matching relies on the similarity of the spatial distribution of spots, which then may vary according to experimental gel running conditions and gel scanning. Quantitative differential protein expression can be erroneous when spots representing the same protein are not correctly matched or when spots representing different proteins are mistakenly matched together.

After matching, the statistical data analysis is carried out to find interesting proteins, i.e. those that have been suppressed or are upregulated. Descriptive statistics summarize the values of matched spots that may indicate significant characteristic spots of gel populations. Usually, 2-DE analysis software offers common statistical tests such as Student's *t*-test, or the Mann–Whitney or Kolmogorov test, but they also propose multivariate analysis, clustering tools or neural networks [26] to locate variations in protein expression profiles. The results can then be visualized through histograms, scatter plots, reports or different views of the gels such as gel transparency, overlapping spot contours or 3-D view.

Figure 3 illustrates the analysis reasoning with eight 2-DE images. These are eight gels from smooth muscle cells of rat samples, out of which four are from newborn and four from aged rats [21]. All images have been matched to gel 930018c-w (upper left in Figure 3). Two classes have been defined, each containing four images (marked "Newborn" and "Aged", respectively, in Figure 3). We first select all groups of spots in each of the two classes and then produce an Inter-Class Report (detail of Figure 3) that shows the Maximum value of each class computed on the spots' normalized volumes (%VOL). A group is a set of spots that have been matched across all gel images, thus representing the same protein. A Statistical Tests report will then show various statistics about the groups in the two classes, such as Student's *t*-test, or the Wilcoxon or Kolmogorov–Smirnov test. This lets the user select and highlight protein spots that are differentially expressed between the two classes. Figure 3 shows the result of one of these tests, highlighting the group of spots that were ranked highest in term of separability between the two



Inter-Class Histograms (%Vol)
The chosen statistics are Mean (100%) and M.S.D.

Inter-Class Report (%Vol)
The chosen statistics are Mean (100%) and M.S.D.

	Match ID ▲	Max	Aged	Newborn
100	752	0.0319583	0	0.0319583
101	751	0.0224426	0.00558588	0.0224426
102	750	3.00485	3.00485	0.216600
103	749	0.0321663	0.0199579	0.0321663
104	748	0.0275269	0.0113875	0.0275269

Statistical Tests (%Vol).

	Match ID	Student T ▲	Wilcoxon	Kolmogorov
1	505	17.9947	0	1.00000
2	750	17.5635	0	1.00000
3	635	14.4208	0	1.00000
4	382	10.8979	0	1.00000
5	846	8.82168	0	1.00000

Figure 3 Gel images analyzed with ImageMaster. See text for details.

classes. In Figure 3, the same group of spots has been highlighted on the Inter-Class Report, the Inter-Class + Intra-Class Histograms and in the gels.

Once specific proteins of interest have been selected via careful data analysis, such as illustrated in Figure 3, further analysis may be carried out to identify or partially characterize those proteins.

3.1.2 **Annotation and Databases**

After extensive analysis of the protein spots has been carried out, including protein identification as detailed in Section 4, 2-DE images may be annotated. The annotations have mainly two functions: linking gels and external databases, and adding information on the gels for later reference. ImageMaster provides broad annotation capabilities in order to include into the gel image all related data and information that has been acquired. Annotations can be added either manually, or they may be imported from an external database, for example through a Laboratory Information Management System (LIMS). Any kind of annotations may be attached to a spot or a pixel, as for example the protein ID, a SWISS-2DPAGE ID (or any other protein database ID), a landmark, calibration values (pI, M_r, intensities) or comments, as well as links to external files such as text files or MS spectra, and also Internet links. By double clicking on the various labels that mark annotated objects, the corresponding piece of information is displayed. In the case of an accession number of a SWISS-2DPAGE entry (e.g. P02990), it launches the default web browser and downloads the corresponding database entry from the local or external user-specified database.

3.2 Analysis of LC-MS Images

So far, we have seen examples from a proteomics workflow in which samples are separated via 2-DE gels. Another possible workflow in proteomics combines separation of proteins and peptides by LC followed by direct analysis by MS (Figure 1). In this case, data may also be represented in two dimensions, the elution time and m/z, and they can be visualized and analyzed as images. LC-MS image analysis systems are still in their infancy. Some prototypes have been presented so far, such as Decyder-MS, MapQuant, SpectroArray, MSD-Viewer and MarkerView, but almost no literature is available describing their performance and characteristics. Pep3D is a tool for producing LC-MS images that is also capable of representing the score values of protein identifications of precursor ions using different color hues [53]. Its simple interface permits the visualization of one experiment at a time and only a single view of the image. The Proteome Informatics Group at the Swiss Institute of Bioinformatics has developed a software tool, MSight [65], for the 2-D representation, visual analysis and comparison of LC-MS datasets. MSight features (i) display and browsing, as an image, of any portion of the collected mass spectra, with a smooth transition from a global overview of all spectra to selected isotopic peaks, (ii) user-friendly navigation through large volumes of data, and (iii) visualization tools to discriminate peptide or protein from noise or to perform differential analysis. This software tool is available free of charge through the ExPASy web server [35] at http://www.expasy.org/MSight. Future versions of this tool will allow for semiautomatic analysis of LC-MS datasets, including quantitative differential proteome analysis.

The images displayed in Figure 4 were obtained from a 42- to 59-kDa fraction of an extract of the human BJAB B cell line. The sample was digested with trypsin and separated by reverse-phase capillary LC coupled to a SCIEX/Applied Biosystems QSTAR quadrupole-TOF mass spectrometer equipped with an ESI source. Spectra were acquired in the m/z range 400–1200, and the image was created using a 0.025 m/z sampling rate and thus contains 55 million measures. The top-right image in Figure 4 displays a 2-D view of the sample described above and the bottom-left image highlights part of this sample in a 3-D view. The bottom right image in Figure 4 shows one single spectrum of the sample.

Image processing of LC-MS datasets can be extremely useful for the monitoring and quality control of experiments as well as knowledge extraction. Software tools specifically developed for the representation of mass spectra along with data from the separation step (such as LC-MS or SDS-MS) provide simple ways to navigate through very large volumes of data. Assessment of the data quality and of the experimental design is simplified by providing a direct means of verifying the quality of the separation and detecting the presence of artifacts and contaminants or mass calibration problems. The redundancy in successive mass spectra may even be used to enhance the signal-to-noise ratio, thereby improving the reliability of MS analysis. Such visual representation of experimental data also helps to understand features such as PTMs of peptides. Most importantly, it allows for automatic or semiautomatic differential proteome analysis by comparing several sets of data, as well as providing fast and intuitive detection of significant qualitative and quantitative differences.

4 Identification and Characterization of Proteins after Separation

Pinpointing differentially expressed protein spots on gels or analyzing mass spectra profiles in LC-MS runs using imaging tools is only the first step in the computer analysis of proteomics data. An important challenge consists of the interpretation of MS and/or MS/MS data to identify (determining the name or sequence of the proteins) as well as to characterize (obtaining information

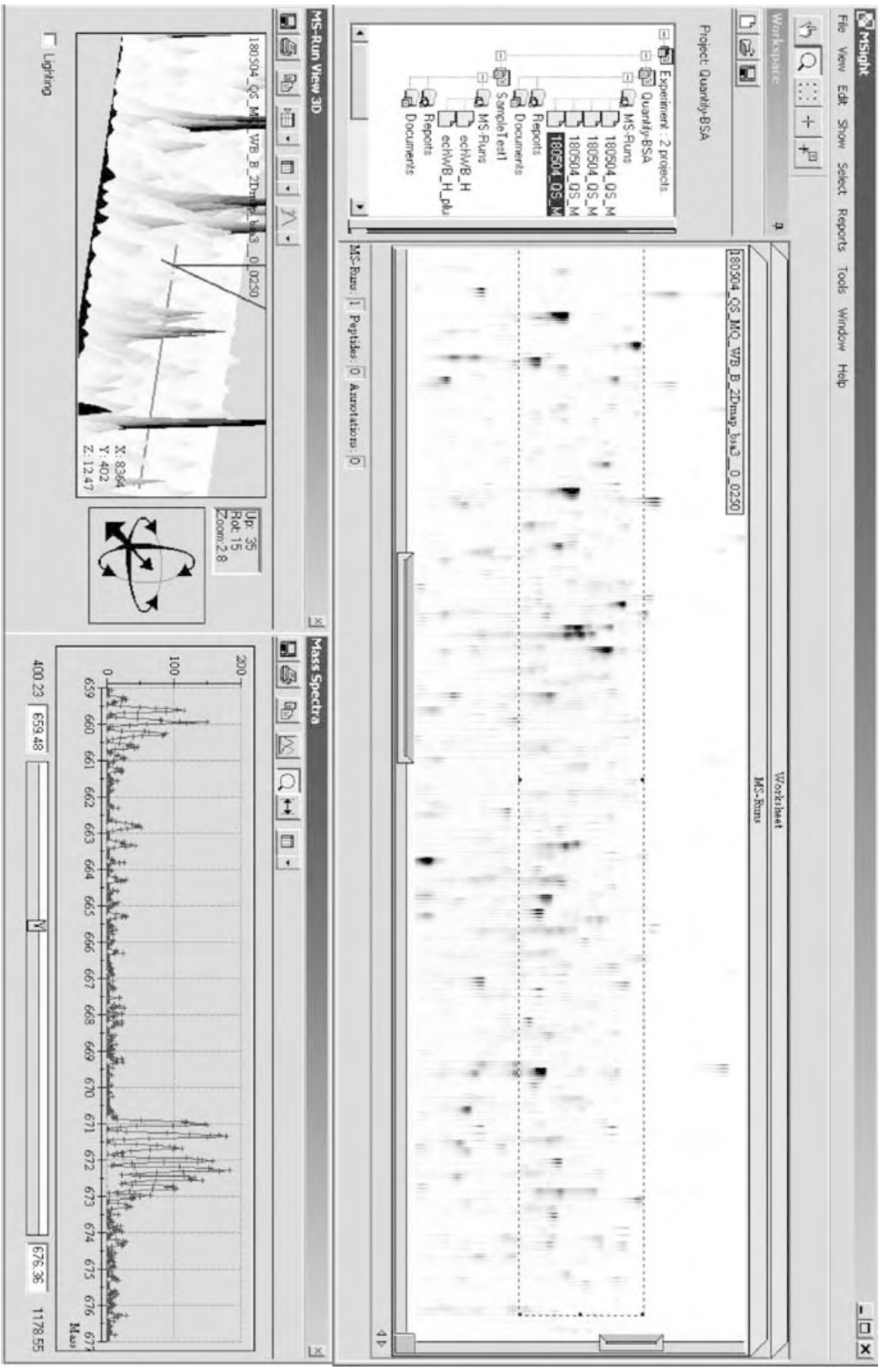


Figure 4 The MSight software showing three different views of the same LC-MS sample: 2-D, 3-D and 1-D (one spectrum) views.

about their function, cellular localization, PTMs, etc.) the concealed proteins. Developing identification and characterization tools from MS data has therefore represented for bioinformatics research a major effort in the last 10 years.

Typically, identification tools identify one or several proteins by matching experimentally obtained protein-related properties against the corresponding theoretical values computed from sequences in a protein sequence database. Once the sequence is known, characterization tools attempt to predict functional and structural features, such as PTMs, splice variants or any other modification or polymorphism.

This section presents the general mechanism of identification and characterization tools from mass spectrometric data, as well as the major tools that are available on the Internet. Most of them are listed on the ExPASy tools page at http://www.expasy.org/tools.

4.1 Identification with MS

MS is typically used to measure the mass of peptides obtained by proteolytic cleavage of one or a mixture of proteins [the peptide mass fingerprinting (PMF) approach] and/or the mass of fragment ions obtained by subsequent fragmentation of one or several peptides [the peptide fragment fingerprinting (PFF) approach]. In PMF, experimental peptide masses are compared to theoretical mass values obtained by applying a proteolytic cleavage rule to the entries in a sequence database. By analogy, PFF compares experimental MS/MS fragmentation mass values of a peptide to theoretical MS/MS spectra computed from the database. In both methods the matches between experimental and theoretical spectra are scored such as to provide a measure of similarity between the two spectra, a higher score indicating a higher likelihood that the corresponding protein or peptide is the target protein or a peptide, respectively, from the target protein.

Software tools to identify proteins using the PMF or PFF approaches share common characteristics:

- A database (typically Swiss-Prot/TrEMBL [12] or NCBI [87]) is searched for the protein, whose theoretical MS (respectively MS/MS spectrum) is most similar to the experimental one.
- A score function measures the similarity between experimental and theoretical spectra; the result is presented as a list of candidate proteins or peptides, sorted by decreasing score values. In PFF, the result shows either candidate peptides or candidate proteins after peptides have been combined to proteins.
- Experimental mass values may be entered manually or copy-pasted. Alternatively, a file with mass values for one or several spectra may be uploaded.

- Experimental parameters may be specified which include the enzyme used for proteolytic digestion as well as the user's confidence in it (maximum number of missed cleavage sites to be considered), the peptide mass tolerance (respectively, precursor ion mass and fragment ion mass tolerances) and various fixed or variable modifications that can affect the protein or peptide sequence, e.g. acetylation, carbamylation or oxidation of residues.
- To increase the specificity and speed of the database search, restrictions may be added. For example, the search may be limited to one species or taxonomic category, or to given p*I* or M_r intervals; resulting protein or peptides may also by filtered out by various other criteria such as the minimum number of matching peaks, minimum score value, etc.

Although PMF can produce excellent results in given situations, particularly when searching databases for species of small and fully sequenced genomes, PFF is a more specific and sensitive identification method and is better adapted when searching in larger databases or when working with complex mixtures of peptides. The sequence information given by the amino acids fragmentation, that are then analyzed using PFF programs, increases the chances of finding true positive hits in database searches. It is possible that even a single peptide (a single MS/MS spectrum) will correctly identify a protein, but this depends on the number of amino acids in its sequence (and as a consequence the peptide coverage on the identified protein). PFF identification is not a perfect method either. Many MS/MS spectra collected during an experiment may not be assigned to any peptide. Possible reasons for these nonmatches can be the presence of contaminants, the poor-quality spectra with noise and unusual fragmentation, spectra derived from proteins not present in the database or with an alternative splicing not annotated in the database, etc. In general, more and more current experimental settings use both PMF and PFF methods in a complementary approach to increase the number of confidently identified proteins.

PMF and PFF are very similar approaches; both correlate experimental spectra with theoretical spectra (respectively, MS and MS/MS spectra). The main difference among the substantial number of existing tools for PMF and PFF lies in their scoring scheme and thus in their ability to identify the correct protein or peptide amongst all candidates, i.e. to distinguish true-positive from false-positive matches. A scoring function must take into account many factors to produce a robust score, like dissimilarities in the peak positions due to internal or calibration errors, peak intensities, noise, contaminants or missing peaks, presence of PTMs, and so on. A variety of different scoring functions have been implemented in various algorithms and programs for PMF and PFF identification [42] such as:

- Aldente [82] (http://www.expasy.org/tools/aldente) is a PMF identification software developed at the Swiss Institute of Bioinformatics. It implements rules, empirical observations and user knowledge in various steps of the identification procedure, and automatically determines the mass deviation of the mass spectrometer by searching for the best mass alignment within the estimated instrument internal precision using a robust method, the Hough transform [73]. Aldente eliminates the need to provide calibrated data, as the alignment procedure considers relative mass variations and thus automatically recalibrates all mass values. The alignment procedure also eliminates outliers thus making the identification much more robust than usual PMF identification programs. Similarly to most other tools on the ExPASy proteomics server (http://www.expasy.org), Aldente takes into account the annotations available in the Swiss-Prot database, particularly PTMs and alternative splicing information. Aldente compares the score of candidate peptides or proteins to the score obtained by matching the spectra to a randomized database in order to statistically compute the likelihood that the candidate proteins or peptides are true positive matches (p-value). Also in line with the other tools on ExPASy, resulting proteins in Aldente may directly be submitted to other ExPASy proteomics tools such as Findpept (that explains peptides that result from unspecific cleavage), PeptideMass (that theoretically cleaves proteins), BioGraph (that graphically represents matched and unmatched peaks in spectra) or FindMod and GlycoMod (see below for details).

- Mascot [66] (http://www.matrixscience.com) includes both a PMF as well as a PFF search tool. Mascot uses a probability-based scoring system that considers matches as random events dependent on the number of entries in the database. Mascot may be obtained from Matrix Science. A free public Internet version is accessible from the above-mentioned URL.

- The ProteinProspector Server [16] (http://prospector.ucsf.edu) from the University of California San Francisco MS facility has a suite of programs that mine sequence databases using MS experiments. Among their tools are MS-Fit for PMF analysis, MS-Tag for PFF, and MS-Seq for a combination of PFF and peptide sequence tag analysis. MS-Seq assumes that short sequence tags are usually easily determined in MS/MS spectra and that their combination with the mass values of the prefix and suffix fragments creating a sequence composed of mass + Tag + mass) increases the discrimination. MS-Seq program is based on the PeptideSearch tool (http://www.narrador.embl-heidelberg.de/GroupPages/Homepage.html) from EMBL [57].

- The PROWL environment [92] (http://prowl.rockefeller.edu) at Rockefeller University comprises ProFound, a PMF identification program which can also be configured to identify components from a mixture of up to four proteins, and PepFrag, an identification tool for PFF. It uses a Bayesian probabilistic score that considers the deviation between theoretical and experimental masses.

- Phenyx (http://www.phenyx-ms.com), developed at Geneva Bioinformatics (GeneBio) in collaboration with the Swiss Institute of Bioinformatics, is a second-generation MS analysis platform that includes both a PMF algorithm using the Aldente alignment algorithm and a PFF identification tool that implements the OLAV true probabilistic scoring scheme [17]. This scoring function, based on a likelihood ratio, takes into account a significant amount of physicochemical parameters of the fragment ions such agmentation pattern probabilities, the presence of different ion series (*a*, *b*, *y*, etc), peak intensities and residue modifications (as annotated in the Swiss-Prot/TrEMBL database). This gives Phenyx the ability to efficiently discriminate between true- and false-positive matches. In addition, the score can be optimized for specific mass spectrometers using a training set of validated identified proteins. Finally, Phenyx also provides a combined PMF and PFF identification mode that makes the identification process more robust. Phenyx, available from GeneBio (http://www.genebio.com), can also be accessed freely at the above-mentioned URL.
- Sequest [28] (http://fields.scripps.edu/sequest/index.html) was the first PFF program to be developed. It compares a simplified experimental spectrum to theoretical spectra built from the databases and ranks candidate peptides using a simple correlation measure. Sequest is available from Thermo Electron Corporation. Examples of tools employing heuristic algorithms include Spectrum Mill, X!Tandem and Sonar [32].

Other tools exist that perform PMF and PFF; however, their intrinsic algorithms are similar to those already cited here. The performances of the above-mentioned tools depend not only on their algorithms, but also on the parameter values and on the content and size of the searched databases. For example, the correct proteins and peptides (the true positives that should be ranked first) would appear at the bottom of the resulting list when the number of sequences in the searched database is very high or when many missed cleavages are allowed. On the other hand, true positives have better chances to appear in the top of the ranked list when restricting the search to the taxonomy of the studied species or group of species. Two metrics are used to compare the performances of MS and MS/MS algorithms and to give some hints on how to determine the confidence level of an identification result. The sensitivity of a tool indicates the ability to make a correct identification regardless of the quality of the data and is calculated by:

Sensitivity = TP/(TP + FN),

where TP stands for true positives and FN for false negatives. The specificity of the tools indicates the ability to calculate low-ranking scores for random (or incorrect) matches and is calculated by:

Specificity = TN/(TN + FN),

where TN stands for true negatives and FP for false positives. In both cases, a certain empirical threshold has to be determined to indicate the correctness of a match. Some of the above-mentioned tools were compared considering these two metrics [15,46]. Regarding the compared PMF tools, ProFound was considered more sensitive and specific than MS-Fit. For the compared PFF tools, Sequest and Spectrum Mill had good sensitivity values, while Mascot, Sonar and X!Tandem had good specificity. It is important to note that these results have to be taken with caution, since the comparisons are made with very specific parameters and databases. In real life, end users submit their MS and MS/MS data to one or two of these tools, and heuristically interpret the different results based on their own biological knowledge, experience and judgment.

Finally, when no matches are found with PMF or PFF, a third alternative approach can be used to obtain protein information from MS/MS data. It is called *de novo* sequencing and consists of inferring the amino acid sequence of the source peptide from an MS/MS spectrum without searching any database. This approach can be useful for determining parts (peptide tags) or all of the sequence of unknown proteins, especially in the case of a proteome from an incompletely sequenced genome, or of mutated or modified proteins that failed to be identified by identification software. Since *de novo* sequencing algorithms do not use database information during spectrum interpretation, they work in a search space composed of the set of all possible sequences that can be represented by the spectrum without any other restriction than the peak patterns. However, these peak patterns can be of very low quality due to fragmentation errors, such as the presence of contaminants and noise, and to missing peptide fragments (noncontinuous signal). Due to these two issues, the size of the search space and the complexity of MS/MS spectra, *de novo* sequencing methods are difficult to automate and current *de novo* software tools often fail to extract correct sequence data.

Despite these issues, some tools are available for *de novo* sequencing. PEAKS [54] is one of them. It uses a dynamic programming algorithm to perform the computation and a mathematical model based on the abundance of ions in the spectra. PEAKS is distributed by Bioinformatics Solutions. Most other software for *de novo* sequencing (such as Spectrum Mill, SeqMS [31], Sherenga [24] or Luthefisk [81]) use a graph theory approach. The spectrum is first translated into a "spectrum graph" where nodes in the graph correspond to peaks in the spectrum and two nodes are connected by an edge if the mass difference between the two corresponding peaks is equal to the mass of an amino acid within a given mass tolerance. The software then attempts to find a path that connects the N- and C-termini, and to connect all the nodes corresponding to the *y* ions (or *b* ions, see Section 2.3 for ion explanations).

Popitam [41], also developed at the Swiss Institute of Bioinformatics, uses a hybrid approach. Similarly to *de novo* sequencing programs, it builds a spectrum graph, but only searches for paths or partial paths (tags) in the graph that match the sequence of candidate peptides from the database. The algorithm specifically allows for identifying modified and/or mutated peptides without any *a priori* knowledge about the expected type of modifications.

4.2 Characterization with MS

Even though the analysis of MS and MS/MS mass spectra can lead to the reliable identification of a protein, a considerable portion of the spectra often remains unmatched and could potentially represent PTMs. Bioinformatics plays an important role in this specific proteomics area with the development of software to help understanding these modifications.

The FindMod tool (http://www.expasy.org/tools/findmod) can be used for *de novo* prediction of PTMs in proteins and potential single-amino-acid substitutions in peptides [88]. It is typically used after PMF identification, and examines mass differences between theoretical peptide masses of a specified known sequence and empirical mass values. If such a mass difference corresponds to a mass difference known to be induced by a PTM, but not yet annotated in the Swiss-Prot entry, FindMod uses a set of rules to predict what amino acid residues in the peptide might carry that modification. FindMod can currently predict 22 modifications including acetylation, methylation, palmitoylation, phosphorylation, etc. For each of these modifications, at least one rule has been established by carefully examining the relevant annotations in the Swiss-Prot and PROSITE [29] databases as well as the literature, with respect to the type of organism, position and amino acid at which the modification can be observed. The same algorithm is applied to suggest possible single-amino-acid substitutions in peptides. In this way, the tool assists in characterizing a gene product and its possible interactions, activities and role in normal and disease states.

Glycosylation is one of the most abundant forms of covalent protein modifications and one of the most complex ones (see also Chapter 45). The range of monomers of which carbohydrate structures in glycopeptides can be composed is broad, and these monomers can be joined in many ways to form linear and branching structures. The diversity of carbohydrate compositions and structures results in a very large number of different potential PTMs that by far exceeds the relatively small number of modifications considered in FindMod. This diversity, as well as a variety of parameters specific to glycosylation, requires a specialized tool, GlycoMod (http://www.expasy.org/tools/glycomod) [18]. MS of glycopeptides can provide useful information on the composition of the attached oligosaccharides: if a peptide is presumed to be glycosylated, the mass of the oligosaccharide can be deduced as the difference between the experimental peptide mass and the mass predicted from the peptide sequence. The potential oligosaccharide compositions corresponding to this mass difference can be calculated from a combination of the masses of possible monosaccharide constituents [64]. GlycoMod considerably facilitates this assignment of mass to composition. The program is linked to Swiss-Prot so that peptide masses of known proteins can be accessed, screened for potential glycosylation sites (both *N*- and *O*-linked) and used in the identification of potential glycopeptides.

Other methods directly predict PTMs from the protein sequence, without any information from mass spectra. Even though they are outside the scope of pure proteomics, they are at least worth to be named. Usually these methods use machine learning or probabilistic approaches to predict modification sites on sequences. The server of the Center for Biological Sequence Analysis (CBS) in Denmark (http:// www.cbs.dtu.dk/services) proposes various predictors based on neural networks. These tools predict specific *N*-linked glycosylation sites in human proteins (NetNGlyc), *O*-glycosylation sites in mammalian (NetOGlyc [45] and YinOYang) and *Dictyostelium discoideum* proteins (DictyOGlyc [38]), acetylation sites (NetAcet [47]), and phosphorylation sites (NetPhos [11]). Myristoylator, a tool developed by the ExPASy team at the Swiss Institute of Bioinformatics, also uses a neural network model to predict the addition of a myristate to a glycine in the N-terminal chain [13]. The Sulfinator on the other hand uses hidden Markov models to localize sulfated tyrosine residues [59].

5 Proteome Databases

Biological related databases can be classified according to the type of information provided, i.e. protein sequences, nucleotide sequences, patterns/profiles, proteomes, 3-D structures, PTMs, genomic and metabolic data. Historically, the expression *proteome databases* was used to only describe databases holding proteomics data, i.e. the data produced by the technologies described in the previous sections, mainly 2-DE gel images and mass spectra. However, according to a more comprehensive definition of current proteomics, this expression has embraced other data resources available to the scientific community. This section briefly describes some of the relevant data resources. A more extensive list is published on a yearly basis by *Nucleic Acids Research*. Here we focus on sequence databases, 2-DE and MS, as well as PTM databases.

5.1 Protein Sequence Databases

The most comprehensive source of protein information is found in protein sequence databases. These can be divided into universal databases, which store protein information from all types of biological sources, and specialized databases, which concentrate their efforts on restricted groups of protein families or organisms. Universal protein sequence databases can be categorized into databases that are simple repositories of sequence data, mostly translated from DNA sequences, and in annotated databases. The latter requires the assistance of curators who screen the original literature, review articles as well as electronic archives. Here we mainly describe Swiss-Prot [12], an annotated universal sequence database, and TrEMBL, an automatically generated sequence database that supplements Swiss-Prot, as well as their integration with other proteomics resources.

Swiss-Prot (http://www.expasy.org/sprot) is a protein sequence database particularly known for its extensive annotation, minimum level of redundancy and maximum level of integration with other databases. Swiss-Prot is mainly manually curated, whereas the vast majority of TrEMBL entries are unannotated or automatically annotated. Created in 1986, release 46.4 includes more than 170 000 entries from more than 9000 different species. Swiss-Prot's main host is the ExPASy server and its eight mirror sites.

Swiss-Prot entries consist of different line types that are grouped into sections. The description section includes, among others, the accession number (a unique entry identifier), the update dates, the protein description (its *name*), the gene name and the taxonomic origin. It is then followed by the reference section, which, for each bibliographical reference, includes the type of experimental work contributing to the entry (sequencing, 3-D structure determination, mutagenesis studies, etc.), the author list and the literature references. The comment section follows then with a variety of textual remarks classified into topics such as function, subunit, similarity, PTM, MS source, etc. The subsequent section is the database cross-reference that provides active links to more than 60 different biological databases. The links to other proteomics databases like SWISS-2DPAGE, PROSITE/InterPro and PDB allow for rapid access to experimental proteomic data, like position and number of protein spots on a 2-DE gel, other members of the same family, or the 3-D structure of the protein. After a keyword section follows the so-called *feature table* section. It describes regions or sites of interest in the sequence as well as documented PTMs, binding sites, active sites, secondary structures, variants, conflicts, etc. The entry ends with the amino acid sequence itself, which is the unprocessed precursor of the protein, before any PTM or processing. Tools that identify proteins from mass spectra should ideally use the information held within the feature table. In order to achieve an optimal approximation of the protein in its mature state, the signal sequence and propeptides should be removed before computing pI and M_r. ExPASy-based proteomics tools such as Aldente and FindMod benefit from the annotation of Swiss-Prot to improve their capacities of identifying and characterizing active chains and proteins annotated with PTMs.

Since its creation, Swiss-Prot has been developed using high-quality manual and computer-assisted annotation, despite the currently large number of genome sequencing projects and, as a consequence the increasing number of sequences that have to be incorporated into Swiss-Prot. This is where TrEMBL (Translation of EMBL Nucleotide Sequence Database [8]) steps in. TrEMBL was created in 1996 as a supplement to Swiss-Prot and consists of computer-annotated entries in Swiss-Prot-like format. It is populated by protein sequences translated from the coding sequences (CDS) in EMBL. In a way, it can be considered as a waiting room to Swiss-Prot; indeed, once annotated, the entries are transferred to Swiss-Prot.

Since 2003, when the maintainers of Swiss-Prot and TrEMBL (the Swiss Institute of Bioinformatics and the European Bioinformatics Institute) joined forces with the PIR group at Georgetown University to form the UniProt Consortium [7], Swiss-Prot and TrEMBL are also known as the “UniProt Knowledgebase”.

As said in the beginning of this section, there are many specialized protein sequence databases available. Their contents vary a lot in terms of range of interest, number of entries, type of information and quality of the data. They are listed and detailed in the special issue on databases of *Nucleic Acids Research* as well as on the ExPASy server (http://www.expasy.org/links.html).

5.2 2-DE Gel Databases

Among proteomics databases, those containing 2-DE gel images with identified proteins, also known as reference maps, are widely used. These databases, usually freely accessible for academics through the Internet, contain clickable maps. The identified spots are linked to their identification method and the description of their identified protein. SWISS-2DPAGE [43] (http://www.expasy.org/ch2d) is the oldest and largest such 2-DE database. Created and maintained at the Swiss Institute of Bioinformatics in collaboration with the University Hospital of Geneva, in August 2006 it contained nearly 40 reference maps of various species including human, mouse, *E. coli*, etc. More than 1300 entries document over 4000 identified "spots". The proteins represented by these spots were identified by matching with other gels, by amino acid composition, by Edman sequencing, by immunoblotting and, mostly, by MS. The text format for each entry is similar to the Swiss-Prot model. It includes specific fields such as the type of master gel from which the protein spot has been identified, the list of gel images associated with the protein entry, as well as other 2-DE specific data, such as the mapping procedure, the spot identifier, the experimental pI and M_r, the MS data, and quantitative data about the protein expression (i.e. physiological and pathological levels, polymorphisms or modifications in specific conditions). The database has cross-references to Swiss-Prot and when no identified spot exists in SWISS-2DPAGE for a given entry in Swiss-Prot, an image is generated highlighting the theoretical position of the corresponding protein. Relevant literature references are provided with links to PubMed. SWISS-2DPAGE data are curated following the Swiss-Prot database standards, i.e. experts manually review the information before making it available. In addition, they follow the MIAPE (Minimum Information About a Proteomics Experiment) guidelines for reporting proteomics experiments recommended by the Human Proteome Organization (HUPO) Proteomics Standards Initiative (PSI) [63]. The MIAPE data exchange model is named as such by analogy to the MIAME (Minimum Information About a Microarray Experiment) model.

The 2-D database of the Max Planck Institute for Infection Biology (http://www.mpiib-berlin.mpg.de/2D-PAGE) is also among the most frequently updated proteomic databases, containing over 20 gels of microbial organisms as well as human, mouse and rat [58]. This database is now part of an interconnected proteome system containing information such as MS spectra, isotope-coded affinity tag (ICAT)-LC-MS spectra, textual descriptions of experimental protocols or results of protein identification [67]. The whole relational database and querying system is implemented in MySQL and uses other open source software such as R for data analysis and graphics. The proteomics local databases are extensively linked to other external public genomic and metabolic databases.

The number of 2-D PAGE databases and related data is slowly but continuously increasing. An up-to-date list can be found in WORLD-2DPAGE (http://www.expasy.org/ch2d/2d-index.html), an index of 2-DE databases and services. More than 25 species are represented in about 300 2-DE maps all over the world. The databases are established in various formats. However, an increasing number follows the principle of federated 2-DE databases [4], according to which the organization of and access to a database must comply with five simple rules. This set of rules was created to homogenize the querying and presentation of such proteomic databases and assist in interconnecting similar data through a cross-reference system, and as a consequence in sharing and distributing 2-DE data in a more effective way. Following those guidelines, the Make2D-DB II package was developed to help research groups to create their own 2-DE databases [60]. This free package not only helps nonexperts to publish their data on the Internet, but it also provides a graphical interface with query capabilities. It can be obtained at http://www.expasy.org/ch2d/make2ddb.html.

5.3 Mass Spectra Repositories

Mass spectra databases are still in their early stages. Three public repositories exist so far. The Open Proteomics database (http://bioinformatics.icmb.-utexas.edu/OPD) is a collection that contains approximately 400 000 spectra representing different experiments from *E. coli*, *Homo sapiens*, *Saccharomyces cerevisiae* and *Mycobacterium smegmatis* [71]. The mzXML Data Repository (http://sashimi.sourceforge.net/index.html) also contains a small number of collections of MS data obtained with different instruments (mainly ThermoFinnigan LCQ and Micromass Q-TOF Ultima) and various mixtures of proteins. The group that maintains this repertoire also distributes tools for MS analysis and has created the mzXML format for the representation of MS data. The third repository, PeptideAtlas (http://www.peptideatlas.org), contains a collection of identified peptides from LC-MS/MS experiments. Currently the experimental results contained in these repositories are not very detailed and data formats are excessively diverse, making their use by other groups difficult.

5.4 PTM Databases

In an era in which more than 100 complete genomes are sequenced per year, the issue of understanding proteins and proteomes relies also on understanding protein modifications that cannot be predicted from the nucleic acid sequences. Most proteins indeed contain PTMs and are not functional unless they are modified. While Swiss-Prot, as a universal database, places a considerable emphasis on the documentation of PTMs within the sequence records, several specialized databases have been set up in recent years to feed this growing field.

RESID [33] is a general database of protein structure modifications (http://www-nbrf.georgetown.edu/pirwww/dbinfo/resid.html), maintained by the National Biomedical Research Foundation in the USA and the PIR group. The database contains descriptive, chemical, structural and bibliographical information on 424 (Release 46.00, June 2006) types of modified amino acid residues. Apart from text-based searches, RESID can also be queried by molecular weight: an average or mono-isotopic mass can be entered (together with a mass variance) to search for all modified amino acid residues in the database with masses similar to the input mass. Unimod (http://www.unimod.org) can be seen as a complementary database to RESID [20]. It is a database for verifiable spectrometric mass values of natural or artificial modifications. It is especially dedicated to mass spectrometric analysis software.

Other databases that are specialized in one particular type of PTM are available. For example, two databases have so far been devoted to glycosylations. The public O-GLYCBASE v6.00 contains about 240 descriptions of glycoproteins that have been experimentally verified to have an O- or C-glycosylation site [37]. O-GLYCBASE entries show the type of *O*-linked sugar involved, the species, the sequence, links to the literature, and cross-references to sequence and structure databases such as Swiss-Prot and PDB. GlycoSuiteDB (http://www.glycosuite.com) is an annotated database of glycan structures restricted to accredited users and submitted to license fees. The database is provided by Proteome Systems, and contains information about most published *O*- and *N*-linked glycans [19]. It is cross-referenced to Swiss-Prot/TrEMBL and it can be queried by mass, by attached protein, by oligosaccharide composition or different modes of textual queries (taxonomy, biological source, etc.).

Protein phosphorylations are currently described in at least three databases differing in the curation levels, details of information and scope of organisms. Phospho.ELM [25] and Phosphorylation site [90] databases (http://phospho.-elm.eu.org and http://vigen.biochem.vt.edu/xpd/xpd.htm, respectively) describe experimentally verified covalent phosphorylations of serine, threonine or tyrosine residues in proteins from eukaryotes (human, mouse, rat and a thousand other organisms) and prokaryotes respectively. PhosphoSite^{TM} (http://www.phosphosite.org) is a curated database dedicated to *in vivo* phosphorylation sites, particularly in human and mouse proteins [44]. Lipoproteins are the object of DOLOP (http://www.mrc-lmb.cam.ac.uk/genomes/dolop), which is restricted to bacterial lipoproteins only. This server also has a predictive algorithm for querying unknown prokaryotic sequences looking for lipoboxes and lists of predicted lipoproteins for multiple completed bacterial genomes [56].

Although the current number of specific PTM databases is still quite small considering the number of known PTMs, it has doubled in the last few years. It is expected that they will multiply because of the increasing amount of data on PTM structures becoming available.

5.5 General Considerations on Databases

It is clear that the databases described above do not cover all the aspects of proteomics. We did not mention databases that use sequence databases to perform calculation and analysis, such as sequence clustering, phylogeny or profile searching, and thus create added-value databases. Other databases report results from functional studies and mutational experiments, or from 3-D structure determination, or describe metabolic pathways. Those were not mentioned here either. It would be impossible to be exhaustive. Some of the databases have already been treated in other chapters. Some of them are permanently updated, some of them have only a short existence, and some of them are not even publicly available. Proteomics databases, as well as data formats, are developed in a dynamic, nonorganized way. To overcome this issue and to facilitate the exchange, dissemination and analysis of the multitude of proteomics data produced by many laboratories, the HUPO PSI has been working on generalized standards representations [63]. Among them are guidelines for reporting proteomics experiments through the MIAPE data integration model, XML formats for microarray and MS data exchange, a list of proteomics ontologies, and guidelines for the comparability of search engine results.

Proteomics information in most databases is accumulated rather than enlarged to a systemic view. The biology understanding is incomplete without quantification and chronology. If the goal of accumulating information is to discover or reveal the function and related biochemical mechanisms, available information has yet to be interconnected, weighed and ordered. Proteome databases are moving from the stage of simple repositories to interconnected systems with intelligent knowledge production means.

6 Conclusion

The development of diagnostic and predictive tools as well as successful therapies for complex polygenic diseases including diabetes, cancer and cardiovascular diseases requires the understanding of the fundamental biological mechanisms implicated in these disorders. This can be achieved under defined environmental conditions with strategies that combine genetic and proteomic tools. Proteome analysis has the ability to detect and identify polypeptides that correlate with disease states, and further lead to the discovery of potential molecular markers and therapeutic targets. Furthermore, proteomic technologies can display the pharmacological and toxic effects of candidate drugs on a disease process. There is a close relationship between drug treatment, protein expression and resulting physiological effects. Most of the time, pharmacological mechanisms entail the secondary regulation or modulation of gene product expressions, in a similar way that complex disease processes alter global protein expression. From this, we can assert that the best drug should be the one that restores global protein expression of a disturbed organism to a normal state. In addition, it is quite unusual that a drug only modulates gene products implied in the disorder. Most of the time, it also causes perturbations in the expression of proteins that are not involved in the disease. This leads to side effects of drugs. Proteomics and bioinformatics are deeply implicated in the understanding of disease and drug effect mechanisms and the design of new drug therapies. This chapter has just given a brief overview of their joined capabilities.

Acknowledgment

The authors would like to acknowledge Dr Manfredo Quadroni (Protein Analysis Facility, Lausanne University, Switzerland) for the LC-MS data.

References

1 Aebersold, R. and M. Mann. 2003. Mass spectrometry-based proteomics. Nature **422**: 198–207.

2 Anderson, L. and J. Seilhamer. 1997. A comparison of selected mRNA and protein abundances in human liver. Electrophoresis **18**: 533–7.

3 Anderson, N. L., J. Taylor, A. E. Scandora, B. P. Coulter and N. G. Anderson. 1981. The TYCHO system for computer analysis of two-dimensional gel electrophoresis patterns. Clin. Chem. **27**: 1807–20.

4 Appel, R. D., A. Bairoch, J. C. Sanchez, J. R. Vargas, O. Golaz, C. Pasquali and D. F. Hochstrasser. 1996. Federated two-dimensional electrophoresis database: a simple means of publishing two-dimensional electrophoresis data. Electrophoresis **17**: 540–6.

5 Appel, R. D., P. M. Palagi, D. Walther, J. R. Vargas, J. C. Sanchez, F. Ravier, C. Pasquali and D. F. Hochstrasser. 1997. Melanie II – a third-generation software package for analysis of two-dimensional electrophoresis images: I. Features and user interface. Electrophoresis **18**: 2724–34.

6 Appel, R. D., J. R. Vargas, P. M. Palagi, D. Walther and D. F. Hochstrasser. 1997. Melanie II – a third-generation software package for analysis of two-dimensional electrophoresis images: II. Algorithms. Electrophoresis **18**: 2735–48.

7 Bairoch, A., R. Apweiler, C. H. Wu, et al. 2005. The Universal Protein Resource (UniProt). Nucleic Acids Res. **33**: D154–9.

8 Baker, W., B. A. van den, E. Camon, P. Hingamp, P. Sterk, G. Stoesser and M. A. Tuli. 2000. The EMBL nucleotide sequence database. Nucleic Acids Res. **28**: 19–23.

9 Barth, H. G., B. E. Boyes and C. Jackson. 1994. Size exclusion chromatography. Anal. Chem. **66**: 595R–620R.

10 Berthod, A. and S. Carda-Broch. 2004. Determination of liquid–liquid

partition coefficients by separation methods. J. Chromatogr. A **1037**: 3–14.

11 Blom, N., S. Gammeltoft and S. Brunak. 1999. Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. J. Mol. Biol. **294**: 1351–62.

12 Boeckmann, B., A. Bairoch, R. Apweiler, et al. 2003. The Swiss-Prot protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res. **31**: 365–70.

13 Bologna, G., C. Yvon, S. Duvaud and A. L. Veuthey. 2004. N-Terminal myristoylation predictions by ensembles of neural networks. Proteomics **4**: 1626–32.

14 Cagney, G., S. Park, C. Chung, B. Tong, C. O'Dushlaine, D. C. Shields and A. Emili. 2005. Human tissue profiling with multidimensional protein identification technology. J. Proteome Res. **4**: 1757–67.

15 Chamrad, D. C., G. Korting, K. Stuhler, H. E. Meyer, J. Klose and M. Bluggel. 2004. Evaluation of algorithms for protein identification from sequence databases using mass spectrometry data. Proteomics **4**: 619–28.

16 Clauser, K. R., P. Baker and A. L. Burlingame. 1999. Role of accurate mass measurement (±10 ppm) in protein identification strategies employing MS or MS/MS and database searching. Anal. Chem. **71**: 2871–82.

17 Colinge, J., A. Masselot, M. Giron, T. Dessingy and J. Magnin. 2003. OLAV: towards high-throughput tandem mass spectrometry data identification. Proteomics **3**: 1454–63.

18 Cooper, C. A., E. Gasteiger and N. H. Packer. 2001. GlycoMod – a software tool for determining glycosylation compositions from mass spectrometric data. Proteomics **1**: 340–9.

19 Cooper, C. A., H. J. Joshi, M. J. Harrison, M. R. Wilkins and N. H. Packer. 2003. GlycoSuiteDB: a curated relational database of glycoprotein glycan structures and their biological sources. 2003 update. Nucleic Acids Res. **31**: 511–3.

20 Creasy, D. M. and J. S. Cottrell. 2004. Unimod: protein modifications for mass spectrometry. Proteomics **4**: 1534–6.

21 Cremona, O., M. Muda, R. D. Appel, S. Frutiger, G. J. Hughes, D. F. Hochstrasser, A. Geinoz and G. Gabbiani. 1995. Differential protein expression in aortic smooth muscle cells cultured from newborn and aged rats. Exp. Cell Res. **217**: 280–7.

22 Cserhati, T. 2002. Mass spectrometric detection in chromatography. Trends and perspectives. Biomed. Chromatogr. **16**: 303–10.

23 Cutler, P., G. Heald, I. R. White and J. Ruan. 2003. A novel approach to spot detection for two-dimensional gel electrophoresis images using pixel value collection. Proteomics **3**: 392–401.

24 Dancik, V., T. A. Addona, K. R. Clauser, J. E. Vath and P. A. Pevzner. 1999. *De novo* peptide sequencing via tandem mass spectrometry. J. Comput. Biol. **6**: 327–42.

25 Diella, F., S. Cameron, C. Gemund, et al. 2004. Phospho.ELM: a database of experimentally verified phosphorylation sites in eukaryotic proteins. BMC Bioinformatics **5**: 79.

26 Dowsey, A. W., M. J. Dunn and G. Z. Yang. 2003. The role of bioinformatics in two-dimensional gel electrophoresis. Proteomics **3**: 1567–96.

27 Eiceman, G. A., J. Gardea-Torresdey, E. Overton, K. Carney and F. Dorman. 2004. Gas chromatography. Anal. Chem. **76**: 3387–94.

28 Eng, J. K., A. L. McCormack and I. J. R. Yates. 1994. An approach to correlate tandem mass spectral data of peptides with amino acid sequences in a protein database. J. Am. Soc. Mass Spectrom. **5**: 976–89.

29 Falquet, L., M. Pagni, P. Bucher, N. Hulo, C. J. Sigrist, K. Hofmann and A. Bairoch. 2002. The PROSITE database, its status in 2002. Nucleic Acids Res. **30**: 235–38.

30 Fenn, J. B., M. Mann, C. K. Meng, S. F. Wong and C. M. Whitehouse. 1989. Electrospray ionization for mass spectrometry of large biomolecules. Science **246**: 64–71.

31 FERNANDEZ-DE-COSSIO, J., J. GONZALEZ, Y. SATOMI, et al. 2000. Automated interpretation of low-energy collision-induced dissociation spectra by SeqMS, a software aid for *de novo* sequencing by tandem mass spectrometry. Electrophoresis **21**: 1694–9.

32 FIELD, H. I., D. FENYO AND R. C. BEAVIS. 2002. RADARS, a bioinformatics solution that automates proteome mass spectral analysis, optimises protein identification, and archives data in a relational database. Proteomics **2**: 36–47.

33 GARAVELLI, J. S., Z. HOU, N. PATTABIRAMAN AND R. M. STEPHENS. 2001. The RESID Database of protein structure modifications and the NRL-3D Sequence-Structure Database. Nucleic Acids Res. **29**: 199–201.

34 GARRELS, J. I. 1989. The QUEST system for quantitative analysis of two-dimensional gels. J. Biol. Chem. **264**: 5269–82.

35 GASTEIGER, E., A. GATTIKER, C. HOOGLAND, I. IVANYI, R. D. APPEL AND A. BAIROCH. 2003. ExPASy: the proteomics server for in-depth protein knowledge and analysis. Nucleic Acids Res. **31**: 3784–8.

36 GÖRG, A., W. POSTEL AND S. GUNTHER. 1988. The current state of two-dimensional electrophoresis with immobilized pH gradients. Electrophoresis **9**: 531–46.

37 GUPTA, R., H. BIRCH, K. RAPACKI, S. BRUNAK AND J. E. HANSEN. 1999. O-GLYCBASE version 4.0: a revised database of *O*-glycosylated proteins. Nucleic Acids Res. **27**: 370–2.

38 GUPTA, R., E. JUNG, A. A. GOOLEY, K. L. WILLIAMS, S. BRUNAK AND J. HANSEN. 1999. Scanning the available *Dictyostelium discoideum* proteome for *O*-linked GlcNAc glycosylation sites using neural networks. Glycobiology **9**: 1009–22.

39 GYGI, S. P., Y. ROCHON, B. R. FRANZA AND R. AEBERSOLD. 1999. Correlation between protein and mRNA abundance in yeast. Mol. Cell. Biol. **19**: 1720–30.

40 HAAB, B. B., M. J. DUNHAM AND P. O. BROWN. 2001. Protein microarrays for highly parallel detection and quantitation of specific proteins and antibodies in complex solutions. Genome Biol **2**: RESEARCH0004.1–13.

41 HERNANDEZ, P., R. GRAS, J. FREY AND R. D. APPEL. 2003. Popitam: towards new heuristic strategies to improve protein identification from tandem mass spectrometry data. Proteomics **3**: 870–8.

42 HERNANDEZ, P., M. MÜLLER AND R. D. APPEL. 2006. Automated protein identification by tandem mass spectrometry: issues and strategies. Mass Spectrom. Rev. **25**: 235–54.

43 HOOGLAND, C., K. MOSTAGUIR, J. C. SANCHEZ, D. F. HOCHSTRASSER AND R. D. APPEL. 2004. SWISS-2DPAGE, ten years later. Proteomics **4**: 2352–6.

44 HORNBECK, P. V., I. CHABRA, J. M. KORNHAUSER, E. SKRZYPEK AND B. ZHANG. 2004. PhosphoSite: a bioinformatics resource dedicated to physiological protein phosphorylation. Proteomics **4**: 1551–61.

45 JULENIUS, K., A. MOLGAARD, R. GUPTA AND S. BRUNAK. 2005. Prediction, conservation analysis, and structural characterization of mammalian mucin-type *O*-glycosylation sites. Glycobiology **15**: 153–64.

46 KAPP, E. A., F. SCHUTZ, L. M. CONNOLLY, et al. 2005. An evaluation, comparison, and accurate benchmarking of several publicly available MS/MS search algorithms: sensitivity and specificity analysis. Proteomics **5**: 3475–90.

47 KIEMER, L., J. D. BENDTSEN AND N. BLOM. 2005. NetAcet: prediction of N-terminal acetylation sites. Bioinformatics **21**: 1269–70.

48 KLOSE, J. 1975. Protein mapping by combined isoelectric focusing and electrophoresis of mouse tissues. A novel approach to testing for induced point mutations in mammals. Humangenetik **26**: 231–43.

49 LACOURSE, W. R. 2002. Column liquid chromatography: equipment and instrumentation. Anal. Chem. **74**: 2813–31.

50 LEE, W. C. AND K. H. LEE. 2004. Applications of affinity chromatography in proteomics. Anal. Biochem. **324**: 1–10.

51 Lemkin, P. F. and L. E. Lipkin. 1981. GELLAB: a computer system for two-dimensional gel electrophoresis analysis. III. Multiple two-dimensional gel analysis. Comput. Biomed. Res. **14**: 407–46.

52 Lescuyer, P., D. F. Hochstrasser and J. C. Sanchez. 2004. Comprehensive proteome analysis by chromatographic protein prefractionation. Electrophoresis **25**: 1125–35.

53 Li, X. J., P. G. Pedrioli, J. Eng, D. Martin, E. C. Yi, H. Lee and R. Aebersold. 2004. A tool to visualize and evaluate data obtained by liquid chromatography–electrospray ionization–mass spectrometry. Anal. Chem. **76**: 3856–60.

54 Ma, B., K. Zhang, C. Hendrie, C. Liang, M. Li, A. Doherty-Kirby and G. Lajoie. 2003. PEAKS: powerful software for peptide *de novo* sequencing by tandem mass spectrometry. Rapid Commun. Mass Spectrom. **17**: 2337–42.

55 MacBeath, G. 2002. Protein microarrays and proteomics. Nat. Genet. **32 (Suppl.)**: 526–32.

56 Madan, B. M. and K. Sankaran. 2002. DOLOP – database of bacterial lipoproteins. Bioinformatics **18**: 641–3.

57 Mann, M. and M. Wilm. 1994. Error-tolerant identification of peptides in sequence databases by peptide sequence tags. Anal. Chem. **66**: 4390–9.

58 Mollenkopf, H. J., P. R. Jungblut, B. Raupach, J. Mattow, S. Lamer, U. Zimny-Arndt, U. E. Schaible and S. H. Kaufmann. 1999. A dynamic two-dimensional polyacrylamide gel electrophoresis database: the mycobacterial proteome via Internet. Electrophoresis **20**: 2172–80.

59 Monigatti, F., E. Gasteiger, A. Bairoch and E. Jung. 2002. The Sulfinator: predicting tyrosine sulfation sites in protein sequences. Bioinformatics **18**: 769–70.

60 Mostaguir, K., C. Hoogland, P. A. Binz and R. D. Appel. 2003. The Make 2D-DB II package: conversion of federated two-dimensional gel electrophoresis databases into a relational format and interconnection of distributed databases. Proteomics **3**: 1441–4.

61 Nikitas, P., A. Pappa-Louisi and K. Papachristos. 2004. Optimisation technique for stepwise gradient elution in reversed-phase liquid chromatography. J. Chromatogr. A **1033**: 283–9.

62 O'Farrell, P. H. 1975. High resolution two-dimensional electrophoresis of proteins. J. Biol. Chem. **250**: 4007–21.

63 Orchard, S., H. Hermjakob, P. A. Binz, C. Hoogland, C. F. Taylor, W. Zhu, R. K. Julian, Jr. and R. Apweiler. 2005. Further steps towards data standardisation: the Proteomic Standards Initiative HUPO 3rd Annual Congress, Beijing 25–27th October, 2004. Proteomics **5**: 337–9.

64 Packer, N. H. and M. J. Harrison. 1998. Glycobiology and proteomics: is mass spectrometry the Holy Grail? Electrophoresis **19**: 1872–82.

65 Palagi, P. M., D. Walther, M. Quadroni, et al. 2005. MSight: an image analysis software for liquid chromatography-mass spectrometry. Proteomics **5**: 2381–4.

66 Perkins, D. N., D. D. J. Pappin, D. M. Creasy and J. S. Cottrell. 1999. Probability-based protein identification by searching sequence databases using mass spectrometry data. Electrophoresis **20**: 3551–67.

67 Pleissner, K. P., T. Eifert, S. Buettner, F. Schmidt, M. Boehme, T. F. Meyer, S. H. Kaufmann and P. R. Jungblut. 2004. Web-accessible proteome databases for microbial research. Proteomics **4**: 1305–13.

68 Pleissner, K. P., F. Hoffmann, K. Kriegel, C. Wenk, S. Wegner, A. Sahlstrom, H. Oswald, H. Alt and E. Fleck. 1999. New algorithmic approaches to protein spot detection and pattern matching in two-dimensional electrophoresis gel databases. Electrophoresis **20**: 755–65.

69 Poetz, O., J. M. Schwenk, S. Kramer, D. Stoll, M. F. Templin and T. O. Joos. 2005. Protein microarrays: catching the proteome. Mech. Ageing Dev. **126**: 161–70.

70 Poole, C. F. 2003. Thin-layer chromatography: challenges and opportunities. J. Chromatogr. A **1000**: 963–84.

71 Prince, J. T., M. W. Carlson, R. Wang, P. Lu and E. M. Marcotte. 2004. The need for a public proteomics repository. Nat. Biotechnol. **22**: 471–2.

72 Regonesi, M. E., F. M. Del, F. Basilico, F. Briani, L. Benazzi, P. Tortora, P. Mauri and G. Deho. 2006. Analysis of the *Escherichia coli* RNA degradosome composition by a proteomic approach. Biochimie **88**: 151–61.

73 Risse, T., L. G. Shapiro, N. Badler, H. Freeman, T. S. Huang and A. Rosenfeld. 1989. Hough transform for line recognition complexity of evidence accumulation and cluster detection. Comput. Vision Graph. Image Process. **46**: 327.

74 Roepstorff, P. and J. Fohlman. 1984. Proposal for a common nomenclature for sequence ions in mass spectra of peptides. Biomed. Mass Spectrom. **11**: 601.

75 Scheele, G. A. 1975. Two-dimensional gel analysis of soluble proteins. Charaterization of guinea pig exocrine pancreatic proteins. J. Biol. Chem. **250**: 5375–85.

76 Sheehan, K. M., V. S. Calvert, et al. 2005. Use of reverse phase protein microarrays and reference standard development for molecular network analysis of metastatic ovarian carcinoma. Mol. Cell Proteomics **4**: 346–55.

77 Smilansky, Z. 2001. Automatic registration for images of two-dimensional protein gels. Electrophoresis **22**: 1616–26.

78 Staby, A., I. H. Jensen and I. Mollerup. 2000. Comparison of chromatographic ion-exchange resins. I. Strong anion-exchange resins. J. Chromatogr. A **897**: 99–111.

79 Steen, H. and M. Mann. 2004. The ABC's (and XYZ's) of peptide sequencing. Nat. Rev. Mol. Cell. Biol. **5**: 699–711.

80 Tanaka, K., H. Waki, Y. Ido, S. Akita, Y. Yoshida and T. Yoshida. 1988. Protein and polymer analyses up tom/z 100 000 by laser ionization time-of-flight mass spectrometry. Rapid Commun. Mass Spectrom. **2**: 151–3.

81 Taylor, J. A. and R. S. Johnson. 1997. Sequence database searches via de novo peptide sequencing by tandem mass spectrometry. Rapid Commun. Mass Spectrom. **11**: 1067–75.

82 Tuloup, M., C. Hernandez, I. Coro, C. Hoogland, P. A. Binz and R. D. Appel. 2003. Aldente and BioGraph: An improved peptide mass fingerprinting protein identification environment. In *Swiss Proteomics Society 2003 Congress: Understanding Biological Systems through Proteomics*. FontisMedia, Lausanne: 174–6.

83 Ünlü, M., M. E. Morgan and J. S. Minden. 1997. Difference gel electrophoresis: a single gel method for detecting changes in protein extracts. Electrophoresis **18**: 2071–7.

84 Vo, K. P., M. J. Miller, E. P. Geiduschek, C. Nielsen, A. Olson and N. H. Xuong. 1981. Computer analysis of two-dimensional gels. Anal. Biochem. **112**: 258–71.

85 Voordijk, S., D. Walther, G. Bouchet and R. D. Appel. 2003. Image analysis tools in proteomics. In *Encyclopedia of the Human Genome*. Nature Publishing, London: 404.

86 Washburn, M. P., D. Wolters and J. R. Yates, III. 2001. Large-scale analysis of the yeast proteome by multidimensional protein identification technology. Nat. Biotechnol. **19**: 242–47.

87 Wheeler, D. L., D. M. Church, R. Edgar, et al. 2004. Database resources of the National Center for Biotechnology Information: update. Nucleic Acids Res. **32**: D35–40.

88 Wilkins, M. R., E. Gasteiger, A. A. Gooley, et al. 1999. High-throughput mass spectrometric discovery of protein post-translational modifications. J. Mol. Biol. **289**: 645–57.

89 Wilkins, M. R., K. L. Williams, R. D. Appel and D. Hochstrasser. 1997. *Proteome Research: New Frontiers in Functional Genomics*. Springer, Berlin.

90 Wurgler-Murphy, S. M., D. M. King and P. J. Kennelly. 2004. The Phosphorylation Site Database: a guide to the serine-, threonine-, and/or tyrosine-phosphorylated proteins in prokaryotic organisms. Proteomics **4**: 1562–70.

91 Yan, J. X., R. Wait, T. Berkelman, R. A. Harry, J. A. Westbrook, C. H. Wheeler and M. J. Dunn. 2000. A modified silver staining protocol for visualization of proteins compatible with matrix-assisted laser desorption/ionization and electrospray ionization-mass spectrometry. Electrophoresis **21**: 3666–72.

92 Zhang, W. and B. T. Chait. 2000. ProFound: an expert system for protein identification using mass spectrometric peptide mapping information. Anal. Chem. **72**: 2482–9.

93 Zhou, X. W., B. F. Kafsack, R. N. Cole, P. Beckett, R. F. Shen and V. B. Carruthers. 2005. The opportunistic pathogen *Toxoplasma gondii* deploys a diverse legion of invasion and survival proteins. J. Biol. Chem. **280**: 34233–44.

94 Zhu, H., J. F. Klemic, S. Chang, et al. 2000. Analysis of yeast protein kinases using protein chips. Nat. Genet. **26**: 283–89.

95 Zhu, H. and M. Snyder. 2003. Protein chip technology. Curr. Opin. Chem. Biol. **7**: 55–63.

Part 8 Protein Function Prediction

29
Ontologies for Molecular Biology

Chris Wroe and Robert Stevens

1 Introduction

This chapter provides an overview of the application of ontologies within the bioinformatics domain, specifically for the representation of protein function. We will provide a background of the ontology field and its technical basis as well as the current developments. We will then link the features of ontologies with the issues faced in bioinformatics, underscoring how ontologies can help. Having done this, we will provide some case studies pertinent to bioinformatics, in general, and protein function, in particular.

Molecular biology currently lacks the mathematical support prevalent in disciplines such as physics and chemistry. In physics, we have laws based in mathematics that allow us to predict planetary orbits, behavior of waves and particles, etc., from first principles. In molecular biology, we cannot yet take a protein sequence and use the amino acid residues present to calculate the structure, molecular function, biological role or location of that protein. All that can be stated is that: "Sequence is related to molecular function and structure". Using this "law", a biologist must compare a novel protein sequence to others that are already well charaterized. If the uncharaterized sequence is sufficiently similar to a charaterized sequence, then it is inferred that the characteristics of one can be transferred to the uncharaterized protein – hence the sequence similarity search. The characterization of single sequences lies at the heart of most bioinformatics, even the new high-throughput techniques that investigate the modes of action of thousands of proteins per experiment. When performing a sequence similarity search, it is not simply the similarity statistics that determine biological insight into the uncharaterized protein.

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

The bioinformatician uses the knowledge about the proteins already characterized in order to arrive at any insights. Thus, it has been said that biology is a knowledge-based discipline [3].

Much of the community's knowledge is contained within its curated databases. In a database such as UniProt/Swiss-Prot, the protein sequence data itself is a relatively small part of the entry. Most of the entry is taken up by "annotation", which can be considered the knowledge component of the database. This knowledge is usually captured as stylized natural language. Even this form of text provides flexibility in the way information can be expressed. The same function can be described using different terms in different resources or the same term can ambiguously describe different concepts in different resources [6, 12]. This semantic heterogeneity is a perennial problem in integrating bioinformatics resources. Although this style of representation is suitable for human readers, the current representation of the knowledge component is difficult to process by machine. Automated integration of data from different resources is becoming essential because of the following features of bioinformatics data:

- *Large quantity of data.* The genome-sequencing projects now imply that data is being produced at increasingly fast rates; a new sequence is deposited in the public genome database EMBL every 10 s. Microarray experiments measuring gene expression and other high-throughput techniques now imply that other data are also being produced in vast quantity, at petabytes per year [13].

- *Complexity of data.* It is difficult to represent most biological data directly in numeric form. As well as the basic data representation, a characteristic of biological data is the many relationships held by each entity. For instance, any one protein has a sequence, functions, processes in which it acts, locations, a structure, physical interactions it makes, diseases in which it may be implicated and many more relationships. Not all relationships are present in a single resource. Information must be integrated to build a complete picture.

- *Volatility of data.* The content of bioinformatics databases reflects the current knowledge of the community and so is constantly evolving.

- *Distribution of data.* In bioinformatics, each small community of researchers often takes the initiative for publishing their data and analysis tools. Bioinformatics uses over 700 of these data resources and analysis tools found all over the Internet [9]. They often have web interfaces through which biologists enter data for analysis, cut and paste results to new web resources or explore results through rich annotation with cross-links [17].

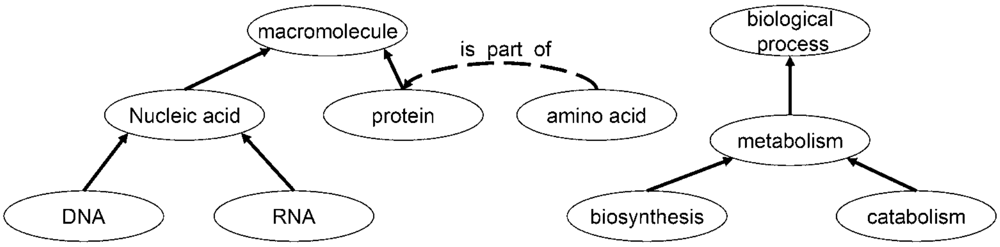


Figure 1 A simple illustration of what can constitute an ontology. Ovals represent concepts, which are classes of instances; the solid arrows represent "is a" relationships, such that all members of a child class are also members of a parent class. The dotted arrow shows a part of relationship between amino acid and protein.

This scene leaves both the maintainers of bioinformatics resources and their users with significant challenges if they are to integrate knowledge from a number of distributed databases. Annotations in each database must be captured in a consistent manner if they are to be comparable to other data. Given the quantity and volatility of the data, it is no longer tenable to manually sift through data for novel insights. Some form of automated assistance is needed to integrate disparate knowledge and filter only relevant data. Therefore, the knowledge component must also be machine interpretable, i.e. computer applications must be able to understand to some extent what the data means: its semantics.

2 Ontologies and their Components

An ontology attempts to capture a community's understanding of a domain as a structured collection of vocabulary terms and definitions [38]. An ontology describes what a community understands and how it communicates about its domain of interest, e.g. the functions of proteins. It describes, in a conceptual form, the things that exist in the domain, both concrete and abstract, such as protein molecule, amino acid, enzyme, α-helix, species, protein function, process, location, disease, etc. It also describes the relationships between these concepts. For example, an ontology can describe knowledge such as the fact that all proteins are comprised of amino acids. Figure 1 shows a simple ontology of some of the basic components of molecular biology.

The term ontology has its origins with Aristotle and in the philosophical domain it is the art of describing things that exist in the world. Computer science has taken this term and altered it. In computer science, an ontology is a conceptualization of a domain of interest, rather than a description of reality. Concepts are units of thought that refer to things in the world – protein, function, sequence [27]. Words are symbols that we use to communicate about things in the world and, in an ontology, terms are used to label concepts. It is these terms that are used by a community to talk about the domain of interest. If the conceptual model of the world (ontology) and the terms for those concepts (lexicon) can be agreed upon by the community, then ambiguity in communication can be reduced. This shared or common understanding of a domain is one of the primary aims of an ontology. A goal of computer science research into ontology representation is to make these conceptualizations of human knowledge processable by computers in a manner that enables them to make inferences about the knowledge stored in them. In summary, the main components of an ontology are:

- The concepts representing entities that exist in the domain. A concept can either be a class that represents a set of instances or a particular instance itself.

- The terms or symbols that label those concepts and allow humans to communicate about those concept or entities in the world.

- The relationships between those concepts. The principle among these is the "is a" relationship that describes a parent–child relationship, or class–subclass, where every instance of the child concept is also a kind of the parent concept. The second major relationship is the "part of" relationship that describes parts and their wholes [39], such as parts of proteins (active site, α-helix, amino acid residue) and their relationship to the whole protein. Other associative relationships are used: causative, nominative, etc.

- Other statements about the concepts and relationships. In logic systems, for example, it is possible to say that sibling concepts are disjoint – it is not possible for an instance to be a member of both classes. Other statements might include equivalence between classes or that the members of child classes completely covers all the members of the parent class.

Although biologists may not have used the term "ontology", the use of classification and description as a technique for collecting, representing and using biological knowledge has a long history in the field. For example, the Linnaean classification of species is ubiquitous and the Enzyme Commission has a classification of enzymes by the reaction that they catalyze [16]. Families of proteins are also classified along axes such as function and structural architecture [12]. Over the past years there has been a surge of interest in using ontologies to describe and share biological data, reflecting the surge in size, range and diversity of data, and the need to assemble it from a broad constituency of sources.

2.1 Ontology Representation

The computer science ontology field is a direct descendant of the knowledge representation (KR) and reasoning work done within the artificial intelligence community over the last several decades. There is not sufficient space in this chapter to provide a complete history of KR; however, it is important to understand that a whole range of formalisms have been developed which capture differing degrees and types of expressivity, computability, and satisfiability, and each has applicability within a particular problem space. Two key styles of representation – frame-based and description logics – have relevance to current ontology work and so we will provide a brief definition.

Frame-based systems are most like object-oriented systems and provide a high degree of structure. They are centered around the idea of a frame, or a class, where each frame represents a set of instances of that frame or class. Each frame has associated slots which represent attributes of the frame. Slots are filled by specific values or by other frames. Slots may be of various kinds. So, for example, frames may have an associated "is a" slot, mentioned earlier, which is used to create a taxonomy within the frame system. The "part of" slot, another highly important slot or relationship, may also be represented in a frame-based system. The frame-based representation system is the most widely used of the KR formalisms and has been used extensively within the life sciences community, e.g. in EcoCyc described in Section 3.6.1.

Description logics (DLs) allow ontologies to be built in a very different way from frame-based systems. Rather than making the author build a taxonomy explicitly (an error-prone task for complex ontologies), a DL provides reasoning capabilities, in the form of a classifier, that will build the ontology from smaller conceptual units. These smaller conceptual units provide sufficient description, a concept with one or more associated relationships, so that the DL reasoner can classify the new concept in the proper place within the ontology. DLs have been of significant interest in the last several years and provide the underlying representation for the Web Ontology Language (OWL).

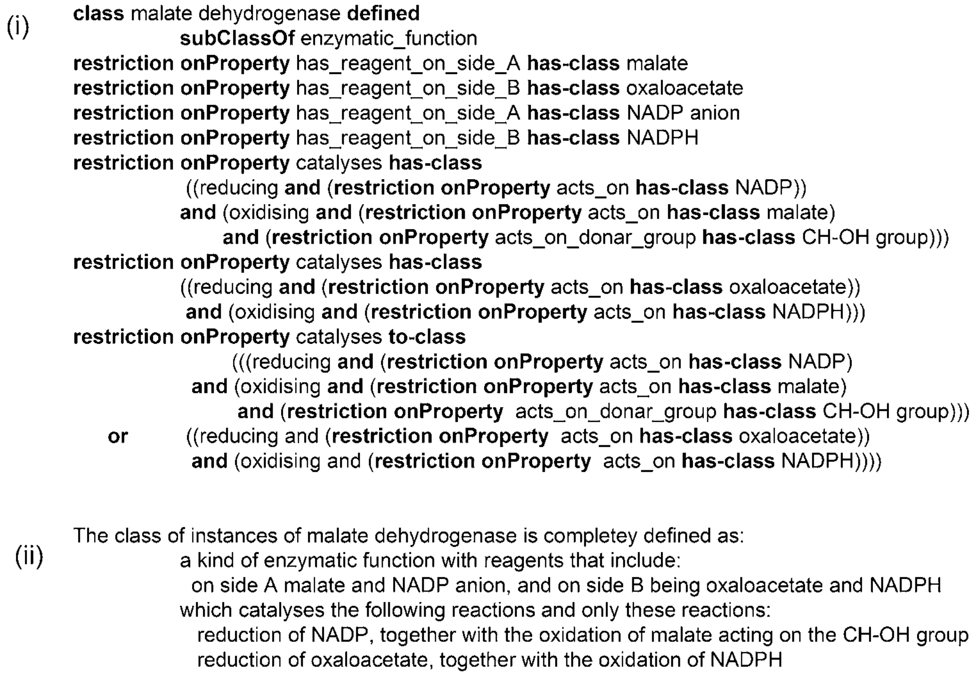
OWL is a KR and transfer language for building ontologies that delivers vocabularies with a formal semantics. It became a W3C recommendation in February 2004 and is descended from the earlier language DAML+OIL [14]. OWL has three increasingly expressive sublanguages: OWL-Lite, OWL-DL, and OWL-Full. It also has a rules language under development for capturing knowledge that cannot be contained in an ontology [1]. The OWL languages are:

- OWL-Lite – provides the capability to describe simple taxonomic classifications and lacks the expressivity to make rich descriptions of classes of instances. It provides a migration path for thesauri and simple taxonomies, such as those commonly seen in current bio-ontologies.

- OWL-DL – an expressive language that is a fragment of first-order logic. This means that it is amenable to machine reasoning. Ontologies described in OWL DL can be checked for logical consistency and subsumption hierarchies (the lattice of "is a" links) inferred from the descriptions of classes formed from the links made between classes [34,35]. This form of OWL will be our focus.
- OWL-Full – more expressive than OWL-DL, but is not yet fully amenable to machine reasoning.

In OWL, classes describe sets of instances or individuals that belong together because they have properties in common (enzyme function is the class of all functionality which can catalyze a reaction – lipase, dehydrogenase, etc.). Classes may be arranged into subsumption hierarchies using the subclass relationship. By stating that receptor binding is a subclass of signal transducer activity, we are stating that all instances of receptor binding are also instances of signal transducer activity. Properties can be used to state relationships between classes and individuals or from individuals to data values. For example, we can say that instances of the class peptide binding act on instances of the class peptide. In OWL DL, we can place restrictions on how properties form relationships that make what that relationship means explicit. We can use existential quantification, to state that all instances of one class have a relationship to some instance of another class, i.e. all peptide binding acts on some peptide (but might also bind something else too), or we can use universal quantification to restrict the target of a relationship to only instances of a certain class – all peptide binding (if it acts on anything) acts on only peptide. We can form more complex expressions by saying that all instances of the class peptide binding acts on some peptide and acts on only some peptide.

OWL-DL is much more expressive than this fragment indicates. For instance, we can describe properties of properties such as transitivity, range and domain constraints, and form hierarchies of properties. It is also possible to say that the instances of two sibling classes do not overlap by using a disjointness axiom – the classes protein and nucleic acid are both kinds of macromolecule, but being disjoint it is not possible for an individual protein to also be a nucleic acid. We can also describe a class as being partial or complete. When the properties of a class are partial, those properties are necessary conditions of class membership, i.e. an instance must have those properties. Describing a class as complete means that an instance having those properties is sufficient to recognize it as a member of that class. So, labeling our class peptide binding as complete would mean that any instance



```
(i)   class malate dehydrogenase defined
                 subClassOf enzymatic_function
      restriction onProperty has_reagent_on_side_A has-class malate
      restriction onProperty has_reagent_on_side_B has-class oxaloacetate
      restriction onProperty has_reagent_on_side_A has-class NADP anion
      restriction onProperty has_reagent_on_side_B has-class NADPH
      restriction onProperty catalyses has-class
                  ((reducing and (restriction onProperty acts_on has-class NADP))
                 and (oxidising and (restriction onProperty acts_on has-class malate)
                     and (restriction onProperty acts_on_donar_group has-class CH-OH group)))
      restriction onProperty catalyses has-class
                 ((reducing and (restriction onProperty acts_on has-class oxaloacetate))
                  and (oxidising and (restriction onProperty acts_on has-class NADPH)))
      restriction onProperty catalyses to-class
                      (((reducing and (restriction onProperty acts_on has-class NADP)
                  and (oxidising and (restriction onProperty acts_on has-class malate)
                      and (restriction onProperty  acts_on_donar_group has-class CH-OH group)))
         or       ((reducing and (restriction onProperty  acts_on has-class oxaloacetate))
                  and (oxidising and (restriction onProperty  acts_on has-class NADPH))))


      The class of instances of malate dehydrogenase is completey defined as:
(ii)             a kind of enzymatic function with reagents that include:
                  on side A malate and NADP anion, and on side B being oxaloacetate and NADPH
                 which catalyses the following reactions and only these reactions:
                   reduction of NADP, together with the oxidation of malate acting on the CH-OH group
                   reduction of oxaloacetate, together with the oxidation of NADPH
```

Figure 2 A complex illustration of the detail an ontology can capture. (i) Human-readable OWL syntax showing the definition for malate dehydrogenase. (ii) A paraphrase of the definition in English

of binding that acts on peptide would have to be a peptide binding. Figure 2 shows how complex an OWL definition can become by showing a candidate definition for the concept malate dehydrogenase in OWL abstract syntax and a paraphrase of the definition in English.

3 Ontologies in the Real World

The previous sections have described the high-level purpose of ontologies and how the ontology language OWL provides us with a wide range of features to formally capture the key concepts of a domain. However, it must be appreciated that building a life science ontology is a significant undertaking. Formal definitions take time to author, with expertise required both in life sciences and KR. Definitions must be checked by the community to ensure they adequately capture their shared understanding. As the communities' knowledge evolves so must the ontology and so at least as much effort is needed to maintain an ontology as was required to build an initial version.

Several approaches have been used by existing ontology developers to mitigate the large amount of effort needed to embark on such a programme.

- *Use of ontology tools.* Sophisticated editing tools can greatly improve the productivity of ontology authors, especially when it becomes necessary to build large ontologies or ontologies with complex concept definitions.
- *Use of existing ontologies where possible.* Having multiple ontologies covering the same areas can be both wasteful and defeat the purpose of a *common* understanding.
- *Incremental development.* By only representing what is currently needed both in terms of coverage and also the degree to which concepts are formally defined, it is possible to greatly reduce development and maintenance effort.

It is important therefore to be aware of current ontology tools, life science ontologies and the way representational features in ontologies support different requirements. Section 3.1 provides an overview of ontology tools and Section 3.2 provides an overview of how the bio-ontology community has begun to support a clearinghouse of community ontologies. To provide both an overview of current ontologies and of phased development, key ontologies will be examined from Section 3.3 onwards to illustrate how the features they include directly support the requirements of that community.

3.1 Ontology Tools

There are now a number of commercial and open source tools for the development, maintenance, merging and visualization of ontologies. A comprehensive survey of ontology tools was conducted in July 2004 by XML.com [7]. We will not attempt to reproduce that survey here, but it is worth mentioning some tools and organizations that are notable within the ontology field.

Undoubtedly, the best-known ontology authoring tool is Protege, from the Stanford Medical Informatics group at Stanford University (http://protege.stanford.edu) [11]. This tool has been in use for over 10 years and it could be argued that it has been around longer than that, since it is an outgrowth of the KR initiatives at Stanford University that have been in existence since the 1970s. Protege has a large and very active user community within a number of commercial and academic projects, and is open source, so that it can be downloaded at no cost and is easily installed. There are several mailing lists to which developers and new users may subscribe, and the tool comes with example ontologies to help new users get started on a project. Protege has a core engine which is extended with plugins. There are a number of plugins and the user community is actively involved with creating new ones. Two recent plug ins are the "OWL-plugin" for writing ontologies in OWL and the "Protege Wizards" plugin developed by the Co-ODE project (http://www.co-ode.org) for simplifying repetitive tasks while writing a large ontology in OWL.

Another tool that is widely used in the life sciences community is DAG-Edit (http://www.geneontology.org/doc/GO.tools.html#dagedit). DAG-Edit was developed at the Berkeley *Drosophila* Genome Project (BDGP) to be used as a part of the knowledge acquisition effort of the Gene Ontology (GO) Consortium. DAG-Edit was initially limited in its representational capabilities, and has been primarily used to represent simple "is a" and "part of" relations; however, it is simple to use, and has been a very effective tool in the development of the very successful GO effort and others. The success of DAG-Edit is in its ability to rapidly assimilate new content for GO.

In addition to these two tools, there are well over 50 other ontology authoring tools, of varying degrees of sophistication and ease of use, from universities and research organizations around the world. There are a large number of commercial organizations that offer commercial ontology authoring tools. Some of the more prominent of these commercial tools are: LinkFactory Workbench from Language and Computing, Integrated Ontology Development Environment (IODE) from Ontology Works, OntoEdit from Ontoprise, OpenCyc Knowledge Server from Cycorp and Construct from Network Inference. Although none of these tools, commercial or academic, yet have complete support for all of the representational capabilities of ontology languages such as OWL or DAML+OIL or the reasoning capabilities of DLs, there are a number of sophisticated tools and a lively marketplace for developing the next breed of tools.

3.2 Bio-ontology Communities

To be effective, an ontology must comprise the shared understanding of a community for a particular subject area. It is therefore important once an ontology has been built to disseminate it so that the effort in building it is not needlessly repeated; the community can contribute to its maintenance and other allied communities can build upon it in their work. The organizations that are promoting the development and the adoption of ontologies in the life sciences field include the GO Consortium, Microarray Gene Expression Data (MGED) group, the Bio-Ontologies Consortium, and the Bio-Pathways Consortium. The GO Consortium (http://www.geneontology.org) brings together 17 model organism databases (at the time of writing) to develop GO. As each new organization joins, they commit to using the GO terms to describe the functionality of gene products in their databases. As a consequence, each new group drives the development of the GO to make available terms needed for that species.

Sequence Ontology (SO) is also part of the GO Consortium. It is a grouping of genome annotation centers, including WormBase, BDGP, FlyBase, the Mouse Genome Informatics group and the Sanger Institute. The aim is to provide a shared vocabulary for the features described on nucleotide and protein sequences. It is intended to range from the basic features seen on a sequence, through interpretations such as "pseudogene" to mutations [8].

The Open Bio-Ontologies effort (OBO) acts as an umbrella under which bio-ontologies may be developed and disseminated (http://obo.sourceforge.net). OBO has a set of principles that govern inclusion:

- Submitted ontologies must be open, but cannot be altered and re-distributed under the same name.
- Use cannot be restricted – ontologies are for sharing.
- A common representation should be used – either the form accepted by DAG-Edit or OWL.
- Ontologies should be nonoverlapping.
- Namespace identifiers should be used in order that any entity within an ontology can be identified as to its source.
- All terms should have a textual definition to prevent ambiguity in interpretation by human readers.

OBO offers access to a wide range of ontologies. Prominent amongst these are several ontologies of anatomy for various species. These are of particular interest to the community as they can be used to identify the biological source of material in experiments – "This microarray experiment used mouse lung", etc. In addition to anatomies, there are also several ontologies of development within species. Finally, there are a growing number of phenotype ontologies available, including traits, disease and behavior.

BioPAX (http://www.biopax.org) is a consortium of pathway databases that aims to develop an exchange language for biological pathways. Pathways include the metabolic, regulatory and signal pathways. The BioPAX initiative aims to overcome the heterogeneity of formats and conceptualizations in the many pathway databases. Initially, BioPAX has used an ontology, written in OWL, to develop a schema for describing the entities and their attributes to be exchanged. Further levels of BioPAX will be developed to provide for controlled vocabularies for the description of pathway data.

MGED has a similar goal to that of BioPAX in that it aims to develop both schema and the vocabularies that fill attributes of that schema for the description of microarray experiments. MGED has been in existence longer than BioPAX, and has a developed an ontology to provide vocabularies for the description of biological samples, their treatments and the experimental conditions pertaining during hybridizations [36]. This ontology is now moving away from the world of model organisms to include toxicology and environmental genomics experiments. As proteomics experimentation develops, there are efforts being made to share descriptions of experiments across broader communities.

While BioPAX, GO and MGED are the most prominent and mature molecular biology ontologies, they are not the only efforts within the life sciences community to develop open-source ontologies. There are also active efforts to develop ontologies in a number of other areas, including:

- Foundational Model of Anatomy (FMA) [31]: an ontology of human anatomy.
- Tissue Ontology [32]: offers a controlled vocabulary for describing tissues across a range of contributing databases.
- Chemical Entities of Biological Interest (ChEBI): a dictionary of small molecular entities that are either products of nature or synthetic products used to intervene in the processes of living organisms (http://www.ebi.ac.uk/chebi/).

All of these organizations have common goals. There is a recognition that this is a community effort and that inclusion of the community will make an ontology work [2,21].

3.3 Incremental Development of Ontologies

Specifications for rich ontology languages such as OWL do not demand that all their features be used. As few features can be used as necessary. Therefore, most successful life science ontology activities narrow the scope and features of the ontology to match their essential set of requirements. They then put in place a procedure to manage the continuing development of the ontology, to ensure it both keeps pace with changing requirements and the changing knowledge of the community [2].

The following section details how a wide range of current bio-ontology-like resources capture different aspects of protein function with different degrees of formality. Their differing design is a result of different requirements. Each resource can be placed on a "feature escalator" as shown in Figure 3. Adding new representational features to the ontology adds more functionality at the cost of complexity to maintain. Fortunately, the decision need not be fixed. Later examples will show how an ontology can be moved along the escalator to meet changing requirements.

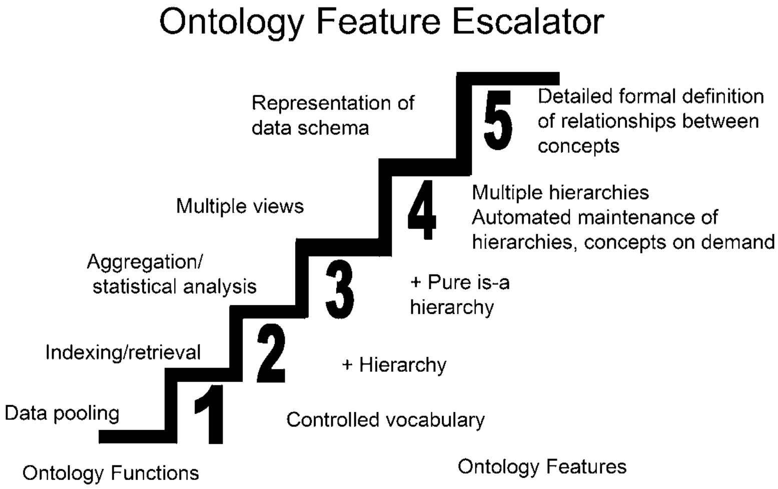

Ontology Feature Escalator
Representation of data schema
5
Detailed formal definition of relationships between concepts
Multiple views
4
Multiple hierarchies
Automated maintenance of hierarchies, concepts on demand
Aggregation/ statistical analysis
3
+ Pure is-a hierarchy
Indexing/retrieval
2
+ Hierarchy
Data pooling
1
Controlled vocabulary
Ontology Functions
Ontology Features

Figure 3 A five-level escalator in which increasing features provide more advanced functionality, but at the cost of complexity.

As stated earlier, the challenges of bioinformatics dictate that some of the key uses of an ontology in life sciences are:

- Pooling knowledge based content between databases by use of a shared vocabulary.
- Supporting database browsing by using the structure of the ontology.
- Aggregating database content by again using the structure of the ontology.
- Supporting the integration or exchange of data by using an ontology to describe its schema.

3.4 Ontology Features to Manage Database Content

3.4.1 A Controlled Vocabulary with Human Readable Definitions

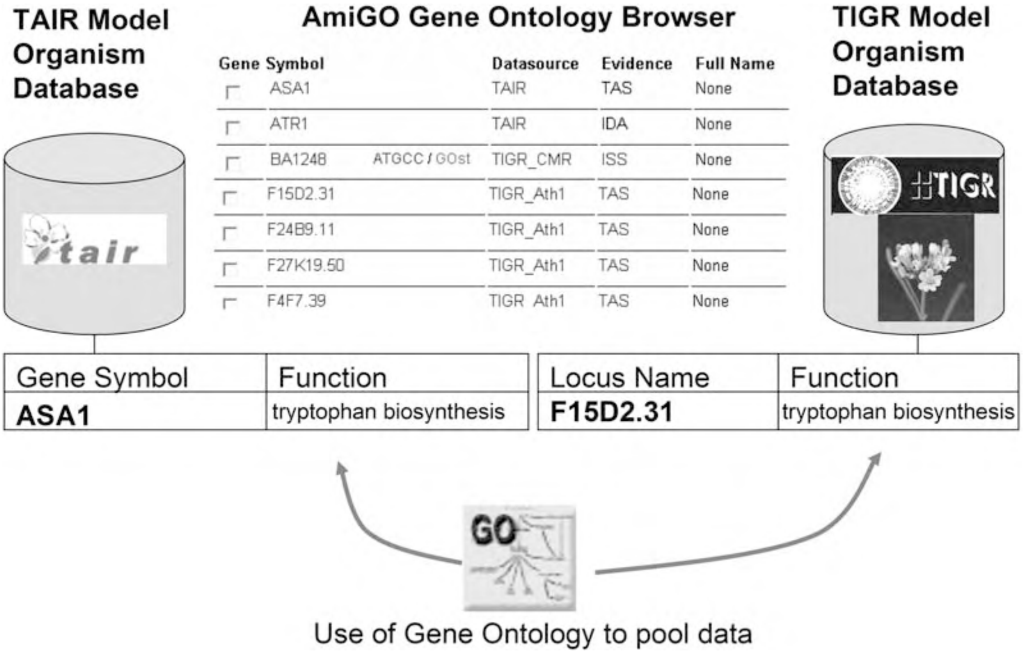
A controlled vocabulary is a constrained list of classes and associated terms used to describe qualitative data. When a community agrees on such a list for aspects of their data, it is possible for computer applications to pool data across distributed databases. This was one of the primary aims of the GO Consortium when developing the GO [37].

3.4.1.1 **Gene Ontology** It is clear that organisms across the spectrum of life, to varying degrees, possess large numbers of gene products with similar sequences and roles. Knowledge about a given gene product (i.e. a biologically active molecule that is the deciphered end-product of the code stored in a gene) can often be determined experimentally or inferred from its similarity to gene products in other organisms. Research into different biological systems uses different organisms that are chosen because they are amenable to advancing these investigations, e.g. the rat is a good model for the study of human heart disease and the fly is a good model to study cellular differentiation. For each of these model systems, there is a database project employing curators who collect and store the body of biological knowledge for that organism. This enormous amount of data can potentially add insight to related molecules found in other organisms. A reliable wet-lab biological experiment performed in one organism can be used to deduce attributes of an analogous (or related) gene product in another organism, thereby reducing the need to reproduce experiments in each individual organism (which would be expensive, time consuming and, in many organisms, technically impossible). However, querying these heterogeneous, independent databases in order to draw these inferences is difficult – the different organism database projects may use different terms to refer to the same concept and the same terms to refer to different concepts. Furthermore, these terms are typically not formally linked with each other in any way. GO seeks to reveal these underlying biological functionalities by providing a controlled vocabulary that can be used to describe gene products and is shared between biological databases. The terminology of GO is used to annotate gene products with respect to three attributes: the specific molecular functions that these products possess, the higher-level biological processes in which they participate and the cellular components in which they can be found. GO has currently been used for over 1 million annotations of gene products within the various participating databases. This facilitates querying for gene products that share biologically meaningful attributes, whether from separate databases or within the same database. Figure 4 shows how it is straightforward to pool data using GO.

GO does not aim to capture a full description of a particular gene product's function. A protein may have a many functions in different circumstances and the exact nature of that functionality may differ subtly in different conditions. Annotation with a GO term simply states that a gene product has been demonstrated or inferred to have a certain capability, without describing in what circumstances.

Appropriate and consistent use of GO concepts requires all annotators to have a common understanding of what each concept represents. Therefore the GO Consortium places a great deal of effort in providing a definition for each concept. The vast majority of GO concepts have a textual definition.

3.4.1.2 **MGED Ontology** When interpreting biological data it is important to consider the experimental conditions under which it was obtained. This is particularly true in microarray experiments where very small variations in conditions or technique lead to incomparable data. As introduced in Sec-


TAIR Model Organism Database
tair
AmiGO Gene Ontology Browser
TIGR Model Organism Database
TIGR
GO
Use of Gene Ontology to pool data

Gene Symbol		Datasource	Evidence	Full Name
ASA1		TAIR	TAS	None
ATR1		TAIR	IDA	None
BA1248	ATGCC / GOst	TIGR_CMR	ISS	None
F15D2.31		TIGR_Ath1	TAS	None
F24B9.11		TIGR_Ath1	TAS	None
F27K19.50		TIGR_Ath1	TAS	None
F4F7.39		TIGR Ath1	TAS	None

Gene Symbol	Function
ASA1	tryptophan biosynthesis

Locus Name	Function
F15D2.31	tryptophan biosynthesis

Figure 4 Two model organism databases, TIGR and TAIR, both use GO concepts to annotate the biological processes in which the gene product has been found to participate. In this case two gene products ASA1 and F15D2.31 are involved in tryptophan biosynthesis. The GO browser AmiGO (http://www.godatabase.org) can quickly access this pooled information.

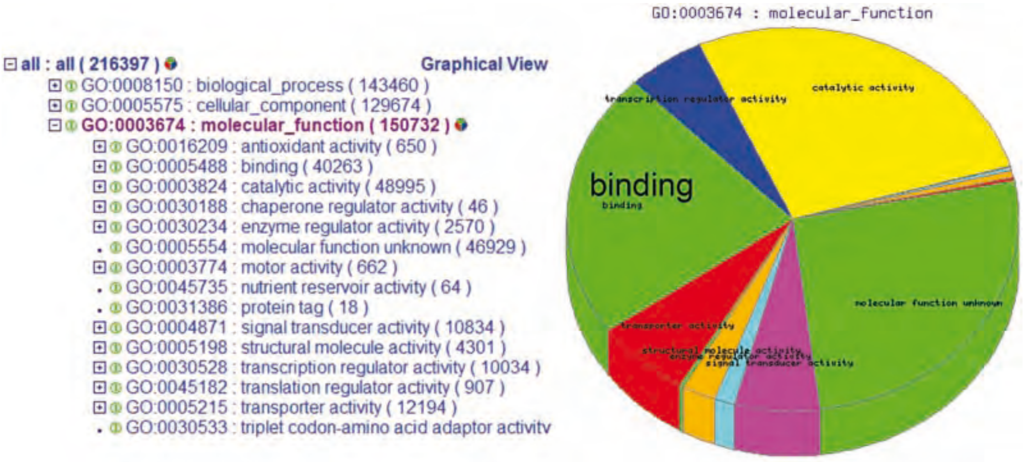
tion 3.2, MGED facilitates the sharing of microarray data generated by functional genomics and proteomics experiments. The main products of MGED are the Minimum Information About a Microarray Experiment (MIAME) guidelines [4] which have been formalized in the Microarray Gene Expression Object Model (MAGE-OM), markup language (MAGE-ML) and associated ontology, the MGED ontology. Information about an array experiment is separated into a number of packages such as array design, experiment and biomaterial. For each package MAGE-OM and MAGE-ML specify the structure of relevant information. For example, that one specific kind of biomaterial "biosource" denotes the initial source of material used in the experiment such as the specific tissue of an organism. The MGED ontology has a structure which mirrors to some extent this organization of packages in MAGE. It has a term that corresponds to "biosource". However, in this case it does not provide terms needed to describe the source in detail, e.g. the species of originating organism. In line with the principles described in Section 3 the annotator is expected to use an existing controlled vocabulary such as the National Center for Biotechnology Information (NCBI) organism taxonomy. In other cases, where an existing vocabulary is not available, the MGED ontology *does* provide a list of controlled terms. There are a list of 51 terms to describe the actions that can be performed on a biomaterial, e.g. “dissect”, “harvest” and “purify”. Each term has an associated text definition to assist annotators in choosing the appropriate term. There is, however, no organization of these lists of terms.

3.4.2 A Structured Controlled Vocabulary

If the number of concepts grows, there is a requirement to organize the concepts into related groups to form a tree, where each concept has a parent. However, the principles by which the concepts are organized can differ greatly. The Medical Subject Headings (MeSH) [22] is an example of a controlled vocabulary in a thesaurus structure used to assign keywords to life science publications. In a thesaurus-like vocabulary the relationship between parent and child concepts is a vague “narrower than, broader than” one. The shape of the tree is designed to (i) assist manual navigation around the tree and (ii) help retrieval of items associated with concepts. A parent–child relationship is added if a search for documents with the parent term should return documents annotated with the child term. Therefore, Accident Prevention G03.850.110.060 is a child of Accidents G03.850.110, despite not being a subclass of the parent term. A computer system can use this structure to support retrieval of data (publications in the case of MeSH).

3.4.3 A Subsumption Hierarchy

Many ontology-like resources have been used, not just for manual navigation and retrieval, but for statistical aggregation of data. For example, GO allows scientists to produce statistics of the number of gene products demonstrating a particular class of function. Figure 5 shows the AmiGO browser (http://www.godatabase.org) with which it is possible to show that currently 27% all annotated gene products demonstrate a binding function (GO:0005488). In order to provide this information a vague “broader than, narrower than” parent–child relationship is not sufficient. In the example above if there is an instance of Accident Prevention it does not hold that it is also an instance of Accidents. If a gene product is annotated with a descendent of binding such as peptide receptor activity (GO:0001653), it must be true that the gene product also shows binding functionality. The GO Consortium call this the “true path rule: the pathway from a child term all the way up to its top-level parent(s) must always be true”. In KR, the relationship in which membership of a child class implies membership of the parent class is called *subsumption*. OWL uses the subclass keyword to denote a subsumption relationship. Limiting the hierarchy to just subsumption relationships, whilst making the structure more amenable to machine processing, makes it more



⊟ all : all (216397)
⊞ GO:0008150 : biological_process (143460)
⊞ GO:0005575 : cellular_component (129674)
⊟ **GO:0003674 : molecular_function (150732)**
⊞ GO:0016209 : antioxidant activity (650)
⊞ GO:0005488 : binding (40263)
⊞ GO:0003824 : catalytic activity (48995)
⊞ GO:0030188 : chaperone regulator activity (46)
⊞ GO:0030234 : enzyme regulator activity (2570)
• GO:0005554 : molecular function unknown (46929)
⊞ GO:0003774 : motor activity (662)
• GO:0045735 : nutrient reservoir activity (64)
• GO:0031386 : protein tag (18)
⊞ GO:0004871 : signal transducer activity (10834)
⊞ GO:0005198 : structural molecule activity (4301)
⊞ GO:0030528 : transcription regulator activity (10034)
⊞ GO:0045182 : translation regulator activity (907)
⊞ GO:0005215 : transporter activity (12194)
• GO:0030533 : triplet codon-amino acid adaptor activitv

Graphical View
GO:0003674 : molecular_function
transcription regulator activity
catalytic activity
binding
binding
molecular function unknown
transporter activity
structural molecule activity
enzyme regulator activity
signal transducer activity

Figure 5 Screen shots from the AmiGO browser (http://www.godatabase.org). The left-hand side shows the subsumption hierarchy for molecular function. The right-hand side shows the aggregated statistics for gene products from all databases annotated with descendants of molecular function terms. In total, 150 732 gene products have been annotated with a molecular function. Of those, 40 263 (27%) have been annotated with binding.

difficult to navigate by users, because seemingly closely related concepts may be in distant parts of the subsumption hierarchy.

GO provides two primary hierarchical relationships that provide structure to the controlled vocabulary, i.e. "is a", and "part of". Although these can both be represented in OWL, or its predecessors, in the spirit of incremental development, the GO Consortium chose the simplest possible representation, a directed acyclic graph, and a simple textual format for its storage, the GO format. "is a" can be interpreted as a subsumption relationship equivalent to the subclass relationship in OWL. "part of" groups structurally related concepts, e.g. components of the nucleus such as nucleolus (GO:0005730), or concepts that are related in terms of subprocesses, e.g. receptor recycling (GO:0001881) is a "part of" (interpreted as "subprocess of") signal transduction (GO:0007165). "part of" is equivalent to an OWL property. Aggregation is still possible using this structure because although it is not true to say that if we have an instance of a nucleolus, we have an instance of its "part of" parent nucleus, it is true to say that a gene product annotated as being localized to the nucleolus belongs to the class of gene products localized to the nucleus.

3.4.4 Multiple Hierarchies

When the nature of concepts in the ontology becomes complex, there are multiple ways in which they can be classified using just subsumption relationships. For example, as shown in Figure 6, the concept adrenocorti-

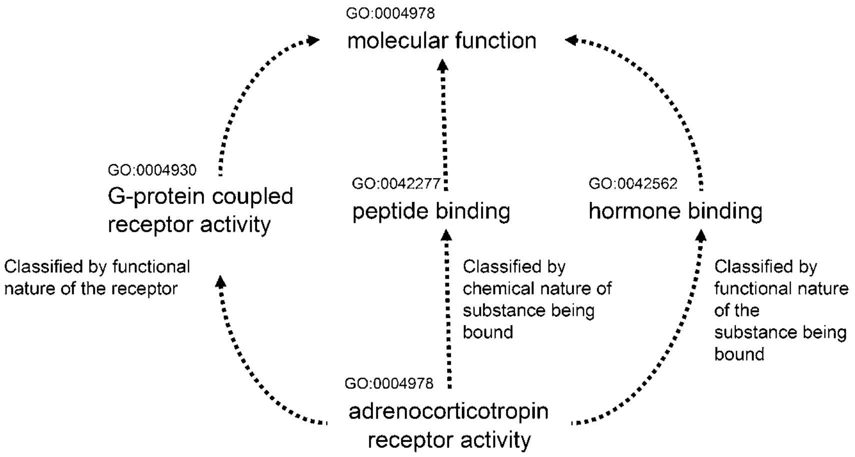

GO:0004978
molecular function
GO:0004930
G-protein coupled receptor activity
GO:0042277
peptide binding
GO:0042562
hormone binding
Classified by functional nature of the receptor
Classified by chemical nature of substance being bound
Classified by functional nature of the substance being bound
GO:0004978
adrenocorticotropin receptor activity

Figure 6 Extract of the multiple classification of adrenocorticotropin receptor activity (GO:0004978) in GO. It is classified in at least three different ways based on the chemical and functional nature of the chemical it binds, and the functional nature of the receptor itself.

cotropin receptor activity (GO:0004978) in GO is classified both in terms of (i) the chemical nature of the substance being bound, e.g., it is a subclass of peptide binding (GO:0042277), (ii) the functional nature of the substance being bound, e.g., it is a subclass of hormone binding (GO:0042562), and (iii) the functional nature of the receptor itself, e.g. G-protein-coupled receptor activity (GO:0004930). Using this more complex structure, protein function annotations can be analyzed along these different axes of classification. The term multiaxial denotes classification structures which simultaneously include different axes of classification. Both OWL and the directed acyclic graph-based GO format allow each concept to have multiple parent concepts and so support multiaxial classification.

This added functionality, however, comes at a cost to the maintainers of the ontology. Maintaining an exhaustive multiaxial hierarchy by hand has been shown to be difficult, leading to a significant rate of omitted parent–child links if the ontology becomes large or the number of parents for each concept becomes significant [30, 40]. Omissions in the classification structure impact on the validity of results from computer applications using that structure.

3.4.5 Formal Definition of Concepts

As mentioned in Section 2, OWL provides many more features that allow ontology authors to capture much more of a concept's definition in a formal manner. DL applications can then interpret these definitions to actually infer multiple subsumption hierarchies automatically. This moves the focus of ontology authors from building the hierarchical structure of the ontology to formally capturing the definitions for each concept.

The GO Next Generation project (GONG) (http://gong.man.ac.uk) demonstrated that, in principle, migrating to a finer grained formal conceptualization in DAML+OIL (and more recently OWL) will allow computation techniques, such as DL to ensure logical consistency, freeing the highly trained curators to focus on capturing biological knowledge [40]. GO is large, so GONG proposed a staged approach in which progressively more semantic information is added *in situ*. DL reasoning is used early and often, and suggested amendments sent to the GO editorial team.

To use the DL to maintain the multiple hierarchy automatically, each GO concept is dissected, explicitly stating the concepts' definition in OWL. This provides the substrate for DL reasoners to infer new and remove redundant subsumption relationships.

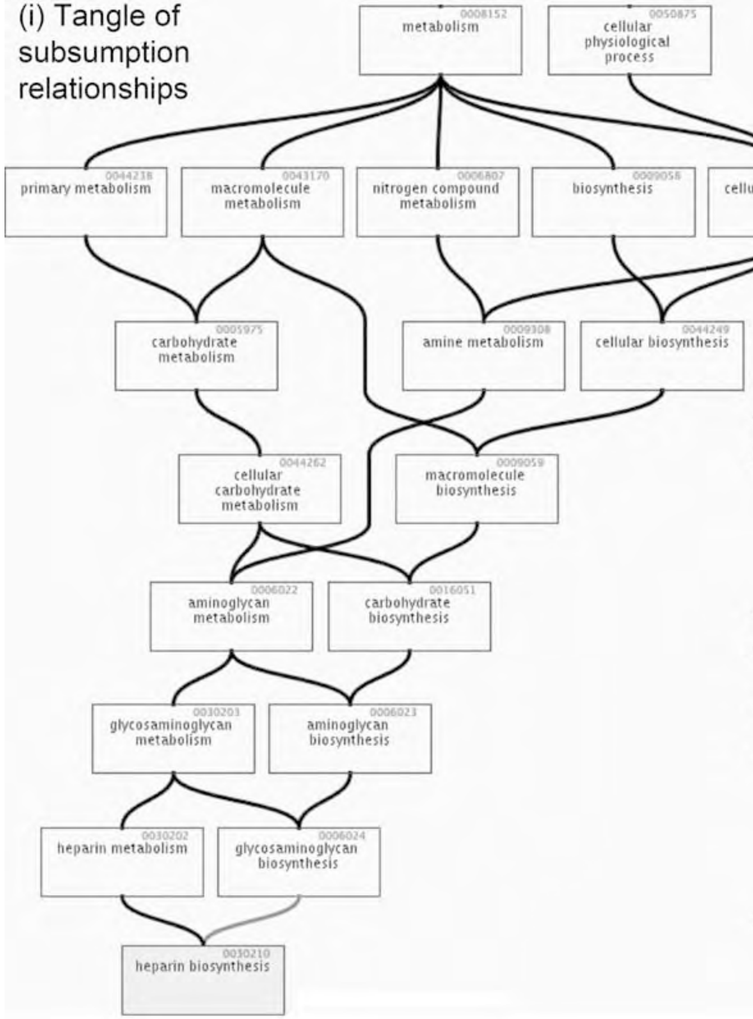
Within a large phrased-based ontology such as GO, which contains many concepts within a narrow semantic range, it is possible to use automated techniques to construct candidate dissections by simply parsing the term name. For example, many metabolism terms in GO follow the pattern "chemical name" followed by either "metabolism", "catabolism" or "biosynthesis". If a term name fits this pattern, a dissection can be created from the relevant phrase constituents as shown in Figure 7. These patterns have to be spotted by a developer and the scripts that generate the DL representation targeted at the appropriate regions of the GO. This provides a semiautomated, targeted approach, which avoids patterns being too general, e.g. confusing "Protein Expression" and "Gene Expression", which may fit a general pattern, but where the former describes a "target" and the latter a "source".

The process of dissection breaks down the existing concept into more atomic concepts that are related in a formal semantic manner. These elemental concepts are then placed in orthogonal taxonomies. Taxonomic information such as the classification of chemical substances which was previously implicit and repeated in many sections of the GO ontology is now made explicit in an independent chemical ontology. The reasoner combines the information in these independent taxonomies to produce a complete and consistent multi-axial classification. The changes reported by the DL reasoner represent mostly additional relationships hard to spot by the human eye and not errors

→

Figure 7 (i) A tangle of subsumption relationships between GO:0030210 heparin biosynthesis and its ancestor concepts as shown in the QuickGO browser (http://www.ebi.ac.uk/ego). The relationship shown in grey has been added by the GONG project. (ii) Two distilled views of this tangle showing ancestors involving more general classes of chemical and more general class of metabolic process. Note that glycosaminoglycan biosynthesis (in grey) should be a parent, but in the version of GO prior to the GONG project there was no subsumption relationship between heparin biosynthesis and glycosaminoglycan biosynthesis. (iii) The formal definitions of these two concepts necessary for the reasoner to infer this subsumption relationship.

(i) Tangle of subsumption relationships



(ii) Distilled views

[*chemical*] biosynthesis (GO:0009058)

View 1: Chemicals

[i] carbohydrate biosynthesis (GO:0016051)

[i] aminoglycan biosynthesis (GO:0006023)

[i] glycosaminoglycan biosynthesis (GO:0006024)

[i] heparin biosynthesis (GO:0030210)

View 2: Process

[i] heparin *metabolism* (GO:0030202)

[i] heparin *biosynthesis* (GO:0030210)

(iii) Example definitions

heparin biosynthesis

```
class heparin biosynthesis defined
    subClassOf biosynthesis
    restriction onProperty acts_on hasClass heparin
                                        (acts_on is unique)
```

Paraphrase: biosynthesis which acts solely on heparin

glycosaminoglycan biosynthesis

```
class glycosaminoglycan biosynthesis defined
    subClassOf biosynthesis
    restriction onProperty acts_on hasClass glycosaminoglycan
```

in biological knowledge. The effect of adding descriptions and using the reasoner can be seen in Figure 7. For example, the reasoner reported that "heparin biosynthesis" has a new "is a" parent "glycosaminoglycan biosynthesis". These reports can then be sent to the editorial team for comment and action if necessary. Even at this preliminary stage of migration, the utility of the approach can be recognized. Many missing and redundant "is a" relationships have been spotted, making GO more complete and robust. Members of the GO editorial team have recognized the potential of using such a logic-based approach to automatically place concepts in the correct location – a task seen as difficult by the team in GO's current hand-crafted form.

The use of reasoning to help maintain the structure of the GO is now being adopted within the GO Consortium [25]. To support the representation of formal definitions and also communities outside the GO Consortium, the language used to represent the GO has been significantly extended to become the Open Bio-ontology Language (OBOL). It now overlaps with the OWL language and it is possible to convert ontologies between these two representations where appropriate. The editing tool DAG-Edit is also being extended to support the formal definition of concepts in this way.

Formal definitions not only support the maintenance of the ontology itself, but also provides other applications with machine-interpretable information for each concept. For example, instead of relying on sequence similarity to retrieve similar proteins, they could be functionally clustered based on their GO functional annotations. This requires several measures of "semantic similarity", e.g. those of Refs. [23, 24] which exploit both the DAG structure of GO and the frequency of use of GO terms within the various databases now annotated with GO. The definition of a metric for "semantic similarity" between GO terms allows us to exploit the machine-interpretable semantics of GO for large data sets. By comparing these metrics to sequence similarity measures we were able to isolate a number of issues in either GO or the use of GO within the annotated databases [23]. We have also investigated the use of these metrics as the basis for a search tool to allow querying within a database (http://gosst.man.ac.uk).

3.5 Ontology Features to Manage Data Schemata

The previous sections examined the ontology features required to represent vocabularies. However, several major life science ontologies describe the *structure* of data rather than specific annotations of its content. Here, the requirements on an ontology are different. Ontologies to describe data schemata are much smaller than vocabulary ontologies. Therefore, less emphasis is needed on paring down their features to the minimum required. In fact, the emphasis is on capturing the relationships between a small number of classes in as much formal detail as possible. The features of the OWL language which can capture the nature of relationships are more often used. For example, the cardinality feature is used to describe the number of instances with which a relationship holds. Domain and range constraints in OWL describe the classes of individuals which can form the source and target of a relationship. Subproperties capture the fact that one relationship can be considered a type of another, e.g. hasStructuralComponent could be a subproperty of hasPart.

The BioPAX ontology is such a case as introduced in Section 3.2. It defines the key entities present in pathway databases and the relationships between them. It complements existing ontologies such as GO. A key class is of course pathway. BioPAX does not itself provide subclasses to describe all the different types of pathway. Instead, it is left to other ontologies to provide this detail. For example, much of the GO biological process taxonomy could be included under pathway. The pathway class has a pathway-component relationship with the class interaction. The interaction class defines a single biochemical interaction with two or more entities. BioPAX does not enumerate a expansive taxonomy of such interactions, which again could be obtained from GO. In this way BioPAX is providing a specification of the relationships between high level classes that span more specific vocabulary ontologies such as GO.

3.5.1 **TAMBIS**

Once an ontology is available that provides a common understanding of multiple overlapping database schemata it can be used to build software applications that guide the scientist in asking a biological question and then transparently translate that question into a number of queries over distributed databases. The TAMBIS (Transparent Access to Multiple Bioinformatics Sources) project built such an ontology-based system [10]. The scientist asks a question by constructing a novel concept using classes and relationships that describes the information of interest. A small sample of such queries are: “Find the active sites of hydrolase enzymes, with protein substrates and metal cofactors” and “Find all chimpanzee proteins similar to human apoptosis proteins”. A concept is a description of a set of instances, so a concept can also be viewed as a query. The TAMBIS system is used for retrieving instances described by concepts in the model, so for example the aforementioned example query could be restated as “Find all instances of the class of chimpanzee proteins similar to human apoptosis proteins”. This contrasts with queries phrased in terms of the structures used to store the data, as in conventional database query environments. This approach allows a biologist to ask complex questions that access and combine data from different sources. However, in TAMBIS, the user does not have to choose the sources, identify the location of the sources, express requests in the language of the source or transfer data items between sources.

The TAMBIS ontology is described using an early DL called GRAIL [28]. The GRAIL representation has a useful extra property in its ability to describe constraints about when relationships are allowed to be formed. For example, it is true that a motif is a component of a biopolymer, but not all motifs are components of all biopolymers. For example, a phosphorylation site can be a component of a protein, but not a component of a nucleic acid, both of which are biopolymers. The constraint mechanism allows the TAMBIS model to capture this distinction and thus only allow the description of biologically meaningful concepts.

3.6 Ontologies for Prediction and Simulation

None of the ontologies described above actually represent in any detail all the communities' knowledge about how a specific protein functions in any one context. Schrager [33] has commented on this shortcoming. Currently, most ontologies only help in the management of information about proteins – pooling information about proteins with similar functions in different species or helping exchange protein function data between databases. Now researchers in the emerging field of systems biology are beginning to look to ontologies to structure much more detailed models at various levels of detail including protein function, to perform predictive analysis or complex simulations of biological systems.

3.6.1 **EcoCyc**

The developers of EcoCyc are a founding member of the BioPAX work group. EcoCyc uses an ontology to describe the richness and complexity of the pathway domain, and the constraints acting within that domain, to specify a database schema [18]. Classes within the ontology form a schema; instances of classes, with values for the attributes, form the facts that with the ontology form the knowledge base. EcoCyc is presented to biologists using an encyclopaedia metaphor. It covers *Escherichia coli* genes, metabolism, regulation and signal transduction, which a biologist can explore and use to visualize information [19].

The instances in the knowledge base currently include 165 Pathways, involving 2604 Reactions, catalyzed by 905 Enzymes and supported by 162 Transporters and other proteins expressed by 4393 Genes [19]. EcoCyc uses the classification of gene product function from Riley [29] as part of this description. Scientists can visualize the layout of genes within the *E. coli* chromosome of an individual biochemical reaction or of a complete biochemical pathway (with compound structures displayed).

EcoCyc's ontology has now been used to form a generic schema MetaCyc, that is used to form the basis for a host of genomic knowledge bases [19].

These ontologies are used to drive pathway prediction tools based upon the genomic information stored in the knowledge base. From the presence of genes and knowledge of their product's function, knowledge can be inferred about the metabolomes of the species in question [20]. Such computations are not only possible with the use of ontology, but EcoCyc's developers would argue that their ontology-based system and the software it supports makes such a complex task easier.

3.7 The Physiome Project

The Physiome project is an international effort to support the computational modeling of the human body, incorporating biochemical, biophysical and anatomical information on cells, tissues and organs [15]. The Physiome Committee of the International Union of Physiological Sciences is encouraging the development of common standards through the use of open markup languages, such as CellML (http://www.cellml.org). Whereas the GO provides a vocabulary to describe the capability of a protein to perform a function, the Physiome project is providing a framework to describe mathematical models of protein function and the contexts within which they apply. Ontologies form a key part of the framework, providing a consistent terminology with which to relate simulation model components with the biological entities described in current bioinformatics databases. They help to describe the context in which a particular model of function operates. The parameters for a particular model may change in specific phenotypes or genotypes. For example, the electrophysiological model of conduction in cardiac cells changes when mutations occur in ion channels within the cell membrane [26]. Capturing these relationships in a consistent manner will become essential if we are to transfer the genetic and proteomic information we gather into an understanding of its impact on a functioning system.

Ontologies are also to be used to relate simulation model components at different spatial scales [5]. For example, a model of tumor growth at the tissue scale will be dependent on the outcome of simulations of the cell cycle within individual cells. Ontologies hold the promise of being an essential *inter lingua* between systems built by diverse communities stretching from biochemists to research clinicians.

4 Summary

Significant progress has been made in the last decade in the creation and adoption of ontologies in the life sciences. There now exist several prominent efforts in the field, the GO Consortium and the wider OBO effort, BioPAX and the MGED Society. These organizations have successfully defined a number of key ontologies in molecular biology and are extending their reach to ontologies in other subdomains within the life sciences. There is strong evidence of the adoption of these ontologies within the community: we have seen GO identifiers used by a number of prominent bioinformatics databases and many of the key conferences in the bioinformatics domain are including tracks on life science ontologies.

In addition to the adoption of ontologies within the biological community, there is also significant work going on in the computer science field to develop richer and more sophisticated ontologies. Adoption of OWL by the W3C is also a significant step in making ontology languages a key building block of IT infrastructure components.

While progress has been substantial, there are still challenges for the broad adoption of biological ontologies in the biological community. The primary obstacles that remain are:

- Adoption of a common language for the representation and exchange of life sciences ontologies.
- Creation of standard ontologies that represent the various key domains of knowledge within the bioinformatics field.
- Improved software tools to assist building ontologies with complex features such as formal definitions, to support the maintenance of ontologies and to support the use of ontologies in applications.

With the recent adoption by the W3C of OWL, OWL has strong support from the Semantic Web community and that community is actively reaching out to life scientists to assist in its adoption. It also has good prospects for continued development as the W3C working group that supports it is very active and comprises some of the top researchers in the field. Most major bio-ontology groups recognize OWL as a standard, including OBO, MGED and BioPAX, so the life science community is well on the way to adopting a common standard.

With respect to standard ontologies, it is unrealistic to expect that there will be a single standard ontology for every domain within the life sciences. However, even convergence on a small number of ontologies within each community is a very positive step forward and would mean progress in the field. The expectation is that leading ontologies such as GO, MGED and others will dominate their field, and will be adopted by the user community as a useful tool.

There is a substantial amount of effort dedicated to improving ontology software tools. Given the standardization and uptake of the OWL language, the hope is that tools will appear not just from within the bioinformatics community, but also from other areas such as the Semantic Web community.

It is heartening to note that bio-ontologies are having an impact within the life sciences community. There is serious work being performed, with good results. Active research underway in the computer science field is planting seeds for the next generation. We can expect bio-ontologies to grow significantly in the years to come.

References

1 ANTONIOU, G. and F. HARMELEN. *A Semantic Web Primer*. 2004. MIT Press, Cambridge, MA.

2 BADA, M., R. STEVENS, C. GOBLE, et al. 2004. A short study on the success of the Gene Ontology. Web Semantics Science, Services and Agents on the World Wide Web **1**: 235–40.

3 BAKER, P., C. GOBLE, S. BECHHOFER, N. PATON, R. STEVENS, and A. BRASS. 1999. An ontology for bioinformatics applications. Bioinformatics, **15**: 510–20.

4 BRAZMA, A., P. HINGAMP, J. QUACKENBUSH, et al. 2001. Minimum information about a microarray experiment (MIAME) toward standards for microarray data. Nat. Genet. **29**: 365–71.

5 CRAMPIN, E. J., M. HALSTEAD, P. HUNTER, P. NIELSEN, D. NOBLE, N. SMITH, and M. TAWHAI. 2004. Computational physiology and the Physiome project. Exp. Physiol. **89**: 1–26.

6 DAVIDSON, S., C. OVERTON, and P. BUNEMAN. 1995. Challenges in integrating biological data sources. J. Comput. Biol. **2**: 557–72.

7 DENNY, M. 2004. Ontology building: a survey of editing tools. XML.com online: http://www.xml.com/pub/a/2004/07/14/onto.html.

8 EILBECK, K., S. LEWIS, C. MUNGALL, M. YANDELL, L. STEIN, R. DURBIN, and M. ASHBURNER. 2005. The sequence ontology: a tool for the unification of genome annotations. Genome Biol. **6**: R44.

9 GALPERIN, M. Y. 2005. The molecular biology database collection: 2005 update. Nucleic Acids Res. **33**: D5–24.

10 GOBLE, C., R. STEVENS, G. NG, S. BECHHOFER, N. PATON, P. BAKER, M. PEIM, and BRASS, A. 2001. Transparent access to multiple bioinformatics information sources. IBM Syst. J. (Special Issue on Deep Computing for the Life Sciences) **40**: 532–52.

11 GROSSO, W. E., H. ERIKSSON, R. W. FERGERSON, J. H. GENNARI, S. W. TU, and M. A. MUSEN. 1999. Knowledge modeling at the millennium (the design and evolution of Protégé-2000). Online at: http://www-smi.stanford.edu/pubs/SMI_Reports/SMI-1999-0801.pdf.

12 HADLEY, C. and D. T. JONES. 1999. A systematic comparison of protein structure classifications: SCOP, CATH and FSSP. Structure **7**: 1099–112.

13 HEY, T. and A. TREFETHEN. 2003. Grid Computing. In: *The Data Deluge: An e-Science Perspective*. Wiley, Hoboken, NJ: 809–24.

14 HORROCKS, I. 2002. DAML+OIL: a description logic for the Semantic Web. Bull. IEEE Comput. Soc. Tech. Comm. Data Eng. **25**: 4–9.

15 HUNTER, P. J. and T. K. BORG. 2003. Integration from proteins to organs: The Physiome Project. Nat. Rev. Mol. Cell Biol. **4**: 237–43.

16 International Union of Biochemistry. 1984. *Enzyme Nomenclature: Recommendations of the Nomenclature Committee of the International Union of Biochemistry on the Nomenclature and Classification of Enzyme-Catalyzed Reactions*. Academic Press (for The International Union of Biochemistry by), Orlando, FL.

17 KARP, P. 1995. A strategy for database interoperation. J. Comput. Biol. **2**: 573–86.

18 KARP, P. and S. PALEY. 1996. Integrated access to metabolic and genomic data. J. Comput. Biol. **3**: 191–212.

19 KARP, P., M. RILEY, M. SAIER, I. PAULSEN, S. PALEY, and A. PELLEGRINI-TOOLE. 2000. The EcoCyc and MetaCyc databases. Nucleic Acids Res. **28**: 56–9.

20 KARP, P., S. PALEY, and P. ROMERO. 2002. The pathway tools software. Bioinformatics **18**: 225S–32.

21 LEWIS, S. 2005. Gene Ontology: looking backwards and forwards. Genome Biol. **6**: 103.

22 LIPSCOMB, C. E. 2000. Medical Subject Headings (MeSH). Bull. Med. Libr. Ass. **88**: 265–6.

23 LORD, P., R. STEVENS, A. BRASS, and C. GOBLE. 2003. Investigating semantic similarity measures across the Gene Ontology: the relationship between sequence and annotation. Bioinformatics **19**: 1275–83.

24 LORD, P., R. STEVENS, A. BRASS, and C. GOBLE. 2003. Semantic similarity measures as tools for exploring the Gene Ontology. Pac. Symp. Biocomput. **8**: 601–12.

25 MUNGALL, C. J. 2004. OBOL: integrating language and meaning in bio-ontologies. Comp. Funct. Genomics **5**: 509–20.

26 NOBLE, D. 2003. Will genomics revolutionise pharmaceutical R&D? Trends Biotechnol. **21**: 333–7.

27 OGDEN C. and I. RICHARDS. 1946. *The Meaning of Meaning*. Harcourt, Brace and World, New York, NY.

28 RECTOR, A., S. BECHHOFER, C. GOBLE, I. HORROCKS, W. A. NOWLAN, and W. D. SOLOMON. 1996. The GRAIL concept modelling language for medical terminology. Artif. Intell. Med. **9**: 139–71.

29 RILEY, M. 1993. Functions of the gene products of *Escherichia coli*. Microbiol. Rev. **57**: 862–952.

30 ROGERS, J., C. PRICE, A. RECTOR, W. SOLOMON, and N. SMEJKO. 1998. Validating clinical terminology structures: integration and cross-validation of Read thesaurus and GALEN. Proc. AMIA Fall Symp., Philadelphia: 845–9.

31 ROSSE, C. and J. L. V. MEJINO. 2003. A reference ontology for bioinformatics: the foundational model of anatomy. J. Biomed. Inf. **36**: 478–500.

32 SCHOMBURG, I., A. CHANG, C. EBELING, M. GREMSE, C. HELDT, G. HUHN, and D. SCHOMBURG. 2004. BRENDA, the enzyme database: updates and major new developments. Nucleic Acids Res. **32**: D431–3.

33 SHRAGER, J. 2003. The fiction of function. Bioinformatics: **19**: 1934–6.

34 STEVENS, R., C. GOBLE, I. HORROCKS, and S. BECHHOFER. 2002. Building a bioinformatics ontology using OIL. IEEE Trans. Inf. Technol. Biomed. **6**: 135–41.

35 STEVENS, R., C. GOBLE, I. HORROCKS, and S. BECHHOFER. 2002. OILing the way to machine understandable bioinformatics resources. IEEE Trans. Inf. Technol. Biomed. **6**: 129–34.

36 STOECKERT, C. and H. PARKINSON. 2003. The MGED ontology: a framework for describing functional genomics experiments. Comp. Funct. Genomics **4**: 127–32.

37 THE GENE ONTOLOGY CONSORTIUM. 2000. Gene Ontology: Tool for the unification of biology. Nat. Gen. **25**: 25–9.

38 USCHOLD, M. and M. GRÜNINGER. 1996. Ontologies: principles, methods and applications. K. Eng. Rev. **11**: 93–136.

39 WINSTON, M., R. CHAFFIN, and D. HERRMANN. 1987. A taxonomy of part-whole relations. Cognitive Sci. **11**: 417–44.

40 WROE, C., R. STEVENS, C. GOBLE, and M. ASHBURNER. 2003. A methodology to migrate the Gene Ontology to a description logic environment using DAML+OIL. Proc. Pac. Symp. Biocomput. **8**: 624–35.

30
Inferring Protein Function from Sequence

Douglas Lee Brutlag

1 Introduction

Inferring gene or protein function from sequence information has been an incredibly valuable approach to interpreting genomes without having to perform detailed biochemical studies on individual gene products. Learning conserved families of proteins and representing conserved functional regions within these families with motifs leverages the known biological attributes of proteins, permitting one to infer proteins belonging to the same family and containing the same conserved functional sites. This procedure can dramatically reduce the amount of experimental work that must be done to characterize the genes in a new genome or gene products in a new proteome. Protein families and motifs also form a much more compact representation of the biological, structural and functional properties of the protein sequence database.

In this chapter we will discuss various methods for representing and learning conserved protein motifs. We will discuss these methods within the standard machine learning paradigms by which motifs can be discovered in a largely unsupervised fashion, being driven primarily by sequence similarity and automatic discovery of protein families and conserved regions. We will also discuss more highly supervised learning of motifs that are driven by finding motifs in smaller subfamilies of proteins defined by biological properties other than sequence.

The simplest way to infer the function of novel proteins is via a pairwise sequence similarity search such as Smith–Waterman local alignment or a BLAST similarity search [1,2] (see also Chapter 3). Pairwise similarity search, in effect, uses every known protein sequence as a pattern to compare with a query protein of unknown function. Sequence-similarity methods require that one find a protein sequence that is closely related to the query protein within a database of proteins with known function. It also requires that one have a database of proteins whose function is known by experiment and not itself inferred from sequence similarity [32]. Otherwise one develops a chain

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

of inferences, each of unknown or dubious significance, often leading to false and unreliable predictions. Other limits on the accuracy of inferring functional or structural information via pairwise alignments have been discussed in depth elsewhere [44, 45, 119, 123].

Another limitation of inferring function from pairwise-similarity search is that such methods usually employ a dynamic programming algorithm that permits arbitrary amino acid substitutions or insertion/deletion gaps (*indels*). If such substitutions or gaps occur within the functional sites of proteins, then the inference of a common function may be in error despite a highly significant overall similarity. The protein in question may, in fact, catalyze a different reaction with a different substrate or cofactor. Even worse, the protein in question may not be functional at all as for pseudogenes. Nevertheless, most users of sequence-similarity search to infer function never check the experimental validity of the target sequence or the impact of the sequence changes between the novel protein and known protein sequence.

A more rigorous method for inferring protein function from sequence is to compare the query sequence with families of sequences (family-wise comparisons) rather than pairwise with individual sequences. The first method for family-wise comparisons was to generate conserved sequence motifs from protein family alignments (Prosite database [15,69,90]). The first advantage of using protein families is that they consist of many proteins that can represent a wide evolutionary or taxonomic range of the functional protein. More importantly, motifs represent the functionally conserved positions in the protein. By learning motifs, one discovers which residues are essential for function and what the range of allowed variation is at each position in the functional site. In this way, the conserved motif (or consensus sequence) is a much better representation of the functional sites. When comparing a sequence motif against a protein of unknown function, only the critical residues are compared. This increases the signal-to-noise ratio and the specificity of the motif, and permits inference to be drawn over a much wider evolutionary or taxonomic range than possible with pairwise sequence comparisons. Conserved motifs also ensure the validity of the functional inference. Insertions, deletions or substitutions in the important conserved residues of the motif will eliminate the inference.

Of course, a major limitation of representing function using sequence motifs is that biological function is not always encoded in a single contiguous series of amino acid residues. Often, functional regions are distributed over several noncontiguous sites in a protein requiring multiple noncontiguous motifs to represent a single biological function (e.g. binding site, interaction site or catalytic site). A good example is the catalytic triad in serine proteases in which each of the three essential residues in the catalytic site is contained in one of three separate conserved motifs (Figure 1) [37, 77, 78]. Another

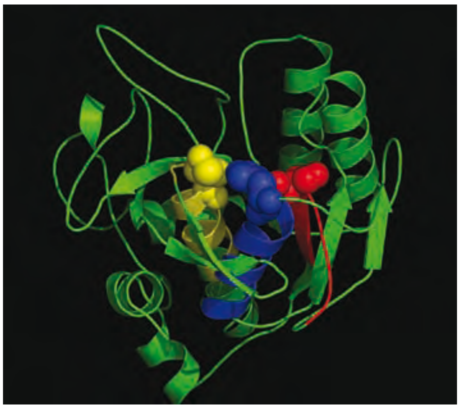


Figure 1 Three conserved motifs containing the three residues of the catalytic triad in the serine protease subtilisin BPN′ of *Bacillus amyloliquefaciens*. The first conserved motif is shown in red (residues 135–146; Prosite PS00136) and contains the active site aspartate (red spheres). The second conserved motif is shown in blue (residues 171–181; Prosite PS00137) and contains the active site histidine (blue spheres). The third motif is shown in yellow (residues 326–336; Prosite PS00138) and contains the active site serine (yellow spheres). Although the three motifs are widely separated in the protein's sequence, the three residues of the catalytic triad are immediately adjacent to each other in the three-dimensional structure.

example is the Walker motifs that encode a nucleotide-binding site (a Rossman fold) and involve two conserved motifs [120]. In such cases one can use a disjunction of motifs or other more sophisticated representations of multiple ordered or unordered sets of motifs.

Another limitation of the motif methods described in this chapter is that biological functions can be conserved without detectable sequence conservation. Examples include conserved chemistry and conserved structure without significant conservation of sequence [14,38,39,52,123]. These problems can be partially addressed by motifs that encode properties of amino acids other than sequence. In the chapter we will discuss network-based motifs (both neural network and Bayesian networks) that can represent patterns of hydrophobicity, charge, volume, contacts, bonding and pairwise correlations. These motifs can represent motifs in a more structural way than sequence based motifs. Other approaches involving structural motifs themselves will not be discussed in this chapter [83,84,107,128].

2 Sequence-based Motif Representations

2.1 Consensus Sequences as Regular Expressions

There are a wide variety of different representations of protein sequence families. As mentioned above, the first method was a consensus sequence or a sequence pattern. Consensus sequences are usually derived from a multiple protein sequence alignment in which the critical functional regions are detected and aligned. The most common representation of the sequence pattern is a regular expression defined by the regular expression syntax such as used in the Unix grep command. The full regular expression syntax can be used to represent protein motifs or one can limit patterns to just fixed-length regular expressions. The full regular expression syntax permits one to represent lists of residues permitted at each position in the consensus sequence as well as variable length regions of "don't care" regions. For fixed-length regular expressions, one has to use multiple regular expressions to represent variable-length sequence motifs.

Regular expressions have some limitations for representing functional motifs. The first is that they are deterministic and discrete. A new sequence either matches or does not match the pattern. Normally there are no approximate matches or concept of suboptimal matches in the regular expression syntax. However, the publicly available program agrep does permit one to search for patterns with a limited number of mismatches [86]. Unfortunately agrep requires one to sacrifice the linear time performance of the grep algorithm making the computation time for pattern search polynomial in terms of the number of variations allowed.

Another limitation of regular expressions is that there are no weights for different positions in the pattern. Each position is weighted equally to the other positions. One can have different mixtures of residues permitted at each position ranging from a single specific amino acid to a "don't care" character and one could assign an information content to each position based on the frequency of residues known at each position; however, apart from such positional composition, each position is weighted equally.

Another problem with consensus sequences prepared manually from protein sequence alignments is they often overfit the existing data. The result of this overfitting is that the motif works well on the existing protein family from which it was derived (the training set), but it does not work nearly as well at inferring function from newly sequenced proteins or proteomes. This overfitting can be evaluated by the cross-validation methods described below.

Finally, the sequence motif pattern, like most of the other motif representation methods discussed below, assumes positional independence of the residues. Sequence variation allowed at one position of a regular expression is not correlated with positional variation at another position. Positional dependencies can arise in a number of ways. First, conserved sequences can represent two or more related sets of proteins that have evolved independently from a common ancestor. These paralogous sets will have their functional positions conserved within each set, but some other positions will have residues that are distinctive for each subset. The distinctive positions in each subset will appear to be correlated with each other. The correlation results from their correlation with their subset.

Another source of covariation between conserved positions could be due to selection for a particular affinity with a substrate, cofactor or other binding site. Most biological interactions are meta-stable so that they can easily associate and dissociate. If a mutation at one position in a binding site over-stabilizes an interaction, this over-stabilization can be compensated for by a change at another position to weaken the interaction. Thus, positional correlations in binding sites may reflect the meta-stable nature of an interaction.

A final source of covariation in protein sequences is due to side-chain–side-chain interactions. These interactions can be van der Waals contacts, electrostatic interactions, charged pairs, hydrogen bonds and general hydrophobic interactions. Changes in one residue in any of these interactions can often be compensated for by a change in another residue, sometimes at great distance in sequence, even though close in three-dimensional space. Positional correlations are often used to detect structural interactions in both proteins and nucleic acids.

One way to represent positional correlation would be to use several motifs to represent subfamilies of proteins within the family. One could use any standard clustering or phylogenetic method to classify all the proteins within a family and then build a separate motif for each subfamily. Such a group of motifs could capture both the conserved and correlated residues in a functional protein motif. One could also apply a minimal length encoding approach to discovering the requisite subfamilies and then obtain a different motif for each subclass. More distant correlated changes often have to be represented by mutual correlations or Bayesian networks or even neural networks.

2.2 Accuracy and Precision of Motifs

Despite their deterministic nature, there are several quantitative measures of how accurate motif representations are. The most important of these measures are specificity, sensitivity and positive predictive value (PPV). The sensitivity of a motif is a measure of what fraction of known members of the functional family are detected by the motif. If a functional family is very diverse or contains multiple subfamilies, one often must construct multiple motifs, one for each subfamily. The sensitivity of each motif is measured relative to its subfamily and the sensitivity of the disjunction of these motifs is measured relative to the entire family.

The specificity of a motif measures the fraction of the nonfunctional proteins that are correctly classified by not having the motif. A more useful measure is (1 – specificity) which measures the number of nonfunctional proteins that do contain the motif. This is a measure of the expected frequency of false-positive prediction by the motif. The expected frequency of false-positive predictions is the fraction of the nonfunctional proteins that contain the motif by chance. The expected frequency of false-positive predictions is usually measured on a protein-by-protein basis; however, it can also be estimated on a residue-by-residue basis. Motifs with an expected frequency of 10^{-3} or less per residue can be used to detect motifs in individual proteins, but are not useful for searching entire proteomes. Highly specific motifs (those with an expected occurrence of false-positive predictions below 10^{-6} per residue) are needed to predict functions in entire bacterial proteomes and even more specific motifs are required for eukaryotic proteomes.

Finally, the PPV of a motif measures what fraction of the sites discovered by the motif are, in fact, correct. The PPV is the ratio of the true sites divided by the true sites plus the false positives.

In order to measure the sensitivity, specificity and positive predictive value of a motif accurately, one must have a "gold standard" set of proteins in which one knows precisely which proteins have the specific function represented by the motif (positive set) and another set of proteins that are known not to have this function (negative set). It is also best if the size of these sets of proteins approximates the expected size of the sets in the sequence databases or proteomes being searched. This usually means that the size of the positive set should be much smaller than that of the negative set. Given these two sets of protein sequences one can apply the motif to both sets and calculate the sensitivity, specificity and PPV as:

$$\begin{aligned}
\text{Sensitivity} &= \text{TP}/(\text{TP} + \text{FN}) \\
&= \text{ fraction of the positive set containing motifs} \\
\text{Specificity} &= \text{TN}/(\text{TN} + \text{FP}) \\
&= \text{ fraction of the negative set not containing motifs} \\
\text{1 - Specificity} &= \text{FP}/(\text{TN} + \text{FP}) \\
&= \text{ expected frequency of false predictions} \\
\text{PPV} &= \text{TP}/(\text{TP} + \text{FP}) \\
&= \text{ fraction of proteins with motifs that are functional,}
\end{aligned}$$

where:

TP = true positives, functional proteins with the motif

FN = false negatives, functional proteins without the motif

FP = false positives, nonfunction proteins with the motif

TN = true negatives, nonfunctional proteins without the motif.

The Prosite database analyzes the occurrences of its motifs in the entire Swiss-Prot database and characterizes each hit as a true-positive or a false-positive hit [17]. In addition, using the known members of a functional family, it determines the number of false negatives and true negatives. From these numbers one could calculate all of the measures described above. However these measures would be an overestimate of the accuracy of the motifs because the Prosite motifs were generated from the sequences in the Swiss-Prot database. This procedure of using the same set for training and evaluation gives rise to resubstitution estimates of sensitivity and specificity which are usually an overestimate of the true sensitivity and specificity of the motifs.

In order to obtain an accurate measure of sensitivity, specificity and PPV for a motif it is essential that the protein set from which the motif is built (training set) be distinct from the protein set on which the motif is tested (test set). One example would be to divide the "gold standard" protein sets into two independent sets – a training set and a test set. Then one would generate motifs from the training set and measure the accuracy on the test set. One could, of course, repeat this training and test procedure reversing the roles of the training set and test sets. This procedure is called cross-validation, and is used to ensure that training and test sets are truly independent. Alternatively, if there is not sufficient data in one-half of the "gold standard" set of functional proteins to build motifs, one can carry out repeated cross-validations using 80 or 90% of the "gold standard" set of proteins and test on the remaining 20 or 10% of the proteins. Repeating this procedure 5 or 10 times so that every protein has served in the test set once gives accurate measures of the sensitivity, specificity and PPV for the motifs.

In addition to these direct measurements of (1 – specificity), the expected frequency of false motifs can also be estimated in a number of different ways. The simplest way to estimate the expected frequency of a motif on a residue basis is by calculating the likelihood of each conserved position in the motif based on amino acid composition alone. This permits one to calculate the expected number of motifs in a protein or a proteome of any length, giving a measure of 1 – specificity of the motif. A more sophisticated method for calculating the expected frequency of a motif in protein sequences which takes into account Markov dependencies of protein sequences and other properties of the regular expression syntax have been developed [8]. In addition to algorithmic methods, empirical methods for estimating the likelihoods of protein motifs have also been generated from random permutations of protein sequence and examining the background frequency of motifs. To get accurate measures of the frequency distribution of motifs it is critical to preserve not only the amino acid composition during the permutation, but also frequencies of peptides of length two and three (the so-called Markov dependencies of length 2 and 3) as well during the permutation process.

2.3 Position-specific Scoring Matrix (PSSM) Motifs

A PSSM is a probabilistic representation of a conserved functional region of a protein [61, 63] (see also Chapter 11). Scoring matrices are generally derived from multiple sequence alignments of proteins generated by a number of alignment algorithms, e.g. ClustalW [118], T-COFFEE [96], DIALIGN [89], PSI-BLAST [2], hidden Markov models (HMMs) [19, 47, 48, 79], etc. (see also Chapter 3). The most highly conserved ungapped regions of the alignments are extracted and referred to as *blocks*. Blocks include contiguous ungapped regions of the protein sequence. These blocks are then converted to probability matrices that give the relative probability of each amino acid in each position in the motif. Finally, the matrix is converted to a log-likelihood form by dividing the probability of each amino acid at each position by the probability of the amino acid in the database of proteins being searched and taking the logarithm of this ratio. If the amino acid is more likely to be found in a particular position in the matrix than in the background distribution of amino acids, then the ratio will be greater than unity and the log-likelihood score will be positive. If the amino acid is less likely to be found at a position in the matrix than the background, then the ratio will be less than unity and the log-likelihood score will be negative. One can then use this log-likelihood scoring matrix to estimate a score for any segment of a newly sequenced protein or proteome by using the protein sequence as an index into the log-likelihood matrix and adding up the likelihood of each residue at each position along the segment. Since we have taken the logarithm of the probabilities, adding the log-likelihood scores is equivalent to multiplying the probabilities at each position. Clearly this method assumes positional independence, just as the regular-expression approach. However, unlike the regular-expression approach, the PSSM approach is probabilistic. The score is a measure of the likelihood that a protein segment belongs to the same functional class as the training set.

2.4 Dirichlet-mixture Prior Probabilities and Pseudocounts

One technical problem occurs when trying to estimate the likelihood of an amino acid when it has never been observed at a position in an alignment or motif. If a particular amino acid has never been observed at a position of a motif, then the frequency and the probability is zero, and one cannot take the logarithm of zero. To circumvent this problem it is common to add a small value to every position in the matrix. For example if one were to add the number 1 to every term in the frequency matrix before taking the log-likelihood ratio, every entry in the matrix would be nonzero. This is conceptually like assuming an observation that contains a uniform distribution of every amino acid at every position. These additions are sometimes referred to as pseudocounts.

Another approach would be to add a small number (not necessarily an integer) proportional to the known amino acid composition of the protein motifs. In this way, the average composition of each column of the matrix would tend toward the average composition of the amino acids. Such probabilities are referred to as Dirichlet-mixture prior probability distributions. The net effect of adding either an integer or a probability distribution to each column of the frequency matrix is to smooth the distribution of amino acids from the observed frequencies towards the average distribution. Hence, these terms are also referred to as smoothing or regularizing parameters.

An important question is how much smoothing is appropriate or, put another way, how many pseudocounts are appropriate when building a scoring matrix. If one has too few protein sequences in the training set, then the matrix may have statistical fluctuations due to small numbers. Under these conditions a large number of pseudocounts would be appropriate. On the other hand, if there are a large number of sequences contributing to the scoring matrix, then very little smoothing may be needed because one already has good estimates of the likelihoods of each amino acid. Basic statistical considerations suggest a minimum of five samples for each probability being estimated which would require 100 sequences in the block to estimate probabilities for 20 amino acids. In fact, scoring matrices are often made for protein families with as few as 10–20 examples. Clearly background smoothing is needed in these cases. Some databases of PSSMs just use a fixed number of pseudocounts to be added to each position (usually 5 or 50); however, it should be clear from the argument above that their should be a variable number of pseudocounts based on the size of the training set. Wu and coworkers [127] demonstrated that the ideal number of pseudocounts can be estimated based on calculating the minimal risk that one would miss a motif. This approach estimates this risk based on the size of the training set and the size of the database that is being searched. By using the number of pseudocounts that minimizes the risk of missing a motif, the overall sensitivity of the motif is maximal.

2.5 Sensitivity and Specificity of PSSM Motifs

Since a PSSM yields a score for every overlapping protein segment to be tested, one must also have a threshold score above which one believes that the segment is an example of a functional motif. How is such a threshold score estimated? Again we usually compare the matrix against a “gold standard” test set of proteins that are known to either contain a functional site or not. We then attempt to find a score that every known site will exceed and all nonfunctional sites will not. However, the scoring matrix will often not cleanly distinguish sites from nonsites perfectly, especially since there are usually many more nonsites than sites. Most often some scores for some nonfunctional sites will exceed the scores for some functional sites. Due to this overlap in the score distributions for sites and nonsites, any given threshold will have some false-positive results and some false-negative results. If one wants to just minimize the number of false calls, one can choose the equivalence point – the threshold score at which there are an equal number of false-positive and false-negative results. If sensitivity (finding all sites) is more critical than getting false positives, then one can lower the threshold score. If specificity is more critical (not getting any false predictions, as in an entire proteome search), then one can increase the threshold score. Hence, the threshold is a parameter that balances sensitivity and specificity.

Because one can vary the score threshold changing the sensitivity and specificity of a scoring matrix, one usually plots the relationship between sensitivity and specificity as the function of the threshold value. Such plots are usually known as receiver operating characteristic (ROC) curves [115]. Each point on an ROC curve represents a different value of the threshold score. See Figure 2 for the definition of the threshold and Figure 3 for an example of an ROC curve for two PSSM motifs. Usually one plots the number of true-positive sites discovered on the y-axis and the number of false positive sites on the x-axis for each threshold. An ideal scoring matrix would only find true sites at high threshold values and then below some score threshold only false positives would appear.

Such an ROC curve would rise vertically up the y-axis to the maximal number of true sites and then turn horizontally as false sites were detected at lower thresholds. More usual ROC curves would show a number of false sites before all of the true sties were found and the curve would fall away from the y-axis. The closer the ROC curve is to the y-axis and the further it is from the x-axis, the more discriminating is the motif. Motifs are often compared by measuring the area under their ROC curves (AUC). The greater this area, the more sensitive and specific the motif is. One can also plot the sensitivity and specificity of discrete motifs (regular expressions) as single points on an ROC curve. If the curve for a PSSM falls underneath this point, then the regular

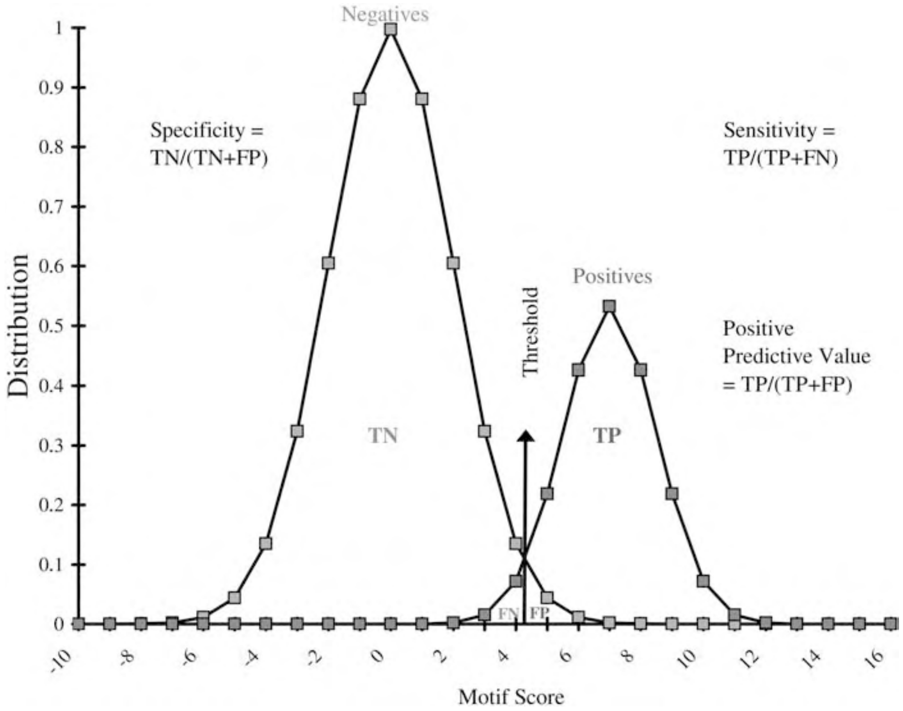


Figure 2 The distribution of motif scores for the positive and negative classes of protein sites often overlap leading to both false-positive and false-negative predictions. The threshold is the value that best separates the positive class from the negative class. Here, we have chosen the equivalence point for the threshold. The equivalence point is the threshold at which the total numbers of false negatives and false positives are equal, and which minimizes the total number of misclassified sites. For some experiments, one wants to maximize the specificity and will raise the threshold value to minimize false predictions. In other experiments, one might want to maximize sensitivity and so will lower the threshold to maximize the number of positives found.

expression is a better motif. If the ROC curve passes over this point, then the matrix is a better motif. An excellent tool for presenting ROC curves is described by Sing and coworkers [104].

Finally, since the size of the true-positive set is very often much smaller than the size of the database as a whole, one often only plots the ROC curve up to a limited number of false positives. One common choice is to plot the ROC curve until the number of false positives equals the number of known true positives in the training set. Another choice would be to plot the ROC curve to some fixed number of false predictions, such as an ROC-50 or an ROC-100 curve [56].

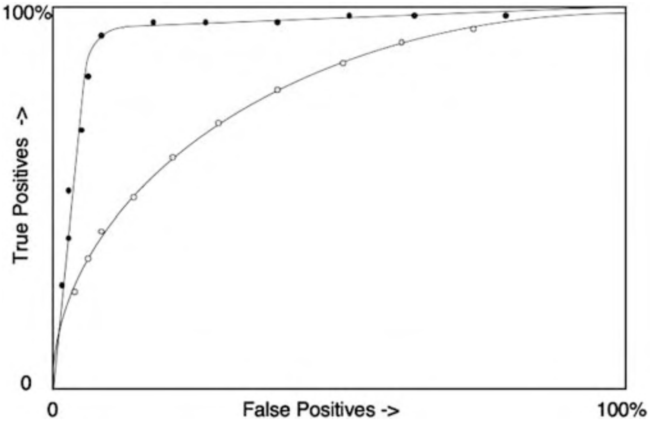


Figure 3 Typical receiver operating characteristic curves for two different PSSM motifs. Each point on each curve represents the fraction of true-positive and false-positive predictions made at a specific threshold (Figure 2). As the threshold score for motif prediction is lowered, the number of true-positive predictions increases for both motifs. However, the motif represented by the upper curve shows more true predictions and fewer false predictions than the lower curve at every threshold. Hence, the AUC for the upper curve is larger than for the lower curve. An ideal motif would have all the true predictions occur before any false predictions and the curve would go from 0,0 to 0,100 along the y-axis and then pass along the top of the graph to 100,100. The area under this ideal motif would be 1.0.

2.6 HMMs

While position specific scoring matrices are excellent for modeling contiguous regions of conserved protein sequence, they cannot model protein regions that have suffered insertion and deletions as can pairwise sequence alignments. A more general representation for conserved proteins that is more global in nature and can model longer conserved regions including regions with variable length insertions and deletions are the HMMs [18, 20, 59, 79]. HMMs were first used to represent protein families. The use of HMMs to represent protein families and protein domains is described in Chapter 3, and one should look there for details on the method. We mention them here only in comparison with other motif representations.

Unlike the PSSMs, HMMs model longer conserved regions that can include multiple motifs, entire protein domains and even represent large protein families [5, 21, 53, 58, 66, 85, 90, 109, 110]. Because of this, they can represent multiple conserved motifs linked by less highly conserved regions. More importantly, many HMM models for protein domains and families can have different likelihoods for insertions and deletions at each position. Such a powerful representation permits a more biological representation of protein domains. For example, in highly structured regions such as α-helices and β-strands, the likelihood of either an insertion or a deletion is generally very low. Insertions or deletions in such structured regions would throw the pattern of hydrophobicity out of register, destroying the structural element. However, in loops or turns in a protein structure, insertions and deletions are more likely. Hence, by examining the probability of insertions and deletions along a protein family represented by an HMM, one can often identify the structured regions and the loops and turns. Even more critically, tight loops (β-turns) can often have insertions but not deletions and these regions can also be recognized.

HMMs are learned from a training set of proteins by an iterative method of repeatedly aligning each sequence in the training set to the model until the model converges. During this process, many different transition probabilities are being estimated at each position, including the likelihood of substitutions, insertions, deletions, extending insertions and extending deletions, terminating an insertion and terminating a deletion. A common HMM will have 25–30 probabilities estimated per position. Statistically one would like at least 150 sequences in the training set in order to estimate all these transition probabilities. However, as with PSSMs, one can use fewer sequences to train an HMM if one has an estimate of the prior probability of the distribution of amino acids. By examining large protein families, a set of distinct types of amino distributions has been calculated referred to as Dirichlet-mixture priors (see Section 2.4) [34, 72, 106]. By using these sets of amino acid distributions that have been seen in previous large HMM models, one can often build an HMM model from fewer than 150 proteins.

Since HMMs are a more general representation of conserved biological functions or structures, they can also be used to represent motifs. Since they are more global in scope, they can also represent structural domains and entire protein families. Since an entire domain often mediates biological function, one can consider protein domain assignment as a critical form of functional assignment. Also, because HMMs are a probabilistic representation, one can derive measures of the likelihood that a novel sequence is a member of a protein domain or family.

2.7 Network Models

All of the motif representations discussed so far have made the assumption of positional independence, i.e. the likelihood of each residue is independent of any of its neighboring residues. Even the HMMs make this assumption in order to be able to use dynamic-programming methods for training and inference. No sequence-based Markov dependencies are permitted. However, we know that this assumption of positional independence is not valid. For example, in order to accurately predict the likelihood of a motif, whether represented by a regular expression or a PSSM, we must use a measure of the Markov dependencies within the background protein database [8, 128]. Accurate estimation of prior probabilities of motifs requires measurement of background Markov dependencies. Also side-chain–side-chain interactions between distant sites in proteins can result in positional correlations.

Studies of many protein families, domains and motifs have shown very important positional dependencies required for structure and function. The most general representation of all positional dependencies would require an estimation of L^2 parameters where $L$is the length of the protein sequence. Generally, one does not have enough examples to estimate this number of parameters so one has to limit one's search for positional dependencies to either local regions or to repetitive regions in a longer structure. Another approach is to look for positional dependencies in reduced representations of the amino acids. Instead of looking for correlations between all 20 amino acids, one can reduce the amino acids into functional or structural subsets and then look for correlations among these properties [74, 75]. For example, if one reduces the amino acids into hydrophobic and hydrophilic subsets, one can discover correlations between these properties along protein sequences. Other properties such as size or volume can lead to discovery of contacts, or charge can lead to discovery of charged pairs or ionic bridges in protein structures. One can also use quantitative methods for search for patterns of hydrophobicity by assigning a measure of hydrophobicity to each residue. Some of the early methods for looking for motifs with positional dependencies involved looking for autocorrelation with protein sequences or performing Fourier analysis of protein sequences. Eisenberg, for example found patterns of hydrophobicity that were typical of α-helices or β-strands in protein sequences [49, 50]. Such patterns of hydrophobicity are extremely useful for inferring protein structure.

Other representations that permit positional dependencies explicitly include network models. Both Bayesian networks and neural networks have been used to represent structural and functional protein motifs. One of the first applications of Bayesian networks to protein structure rediscovered the hydrophobic patterns of helices and strands. Bayesian networks were also used to discover the motif [Phe Xxx Yyy Zzz His] that terminates α-helices. Structural analysis of this so-called C-terminal capping sequence shows that the histidine bends back and interacts with the phenylalanine, terminating the α-helical structure [46, 75]. Bayesian approaches have also been used to represent patterns of amino acids in repeating proteins such as coiled coils or triple helices such as found in many structural proteins (myosin, collagen, etc.) [27–29, 33, 87, 124]. Chapter 35 also discusses Bayesian networks.

2.8 Neural Networks

Neural networks permit the examination of local regions of amino acid sequence for nonlinear combinations of properties other than just the sequence itself. Thus, neural networks can learn patterns of hydrophobicity, charge, hydrogen bonds and other properties in combination with more classical amino acid patterns [92, 93, 98]. Due to their extreme flexibility, neural networks can learn arbitrary quantitative patterns of amino acids that may not be obvious in the sequence itself. Such powerful methods are ideally suited to examining regions of proteins that are associated with membranes or with other proteins and which the association depends more on the physical properties of the protein than its sequence. Also, the extreme flexibility of the networks permits one to encode nearest-neighbor information as well as relationships extending over the entire region of the protein being evaluated. Such methods have been used to learn motifs that can infer protein structural regions such as helices and strands, coding regions, transmembrane regions, signal peptides, post-translation modification signals, nuclear localization and secretion signals, etc. [24, 30, 40, 51, 70, 73, 94, 95].

Discovering and training a neural network takes some effort to determine the factors necessary for recognition. Considerable biological knowledge can be built into the architecture of the network from the start (width, number of hidden layers, parameterization of the sequence, specific sequence patterns, etc.). Some biological insights can also be extracted from the neural network once it is trained as well (internal periodicities, relative importance of specific positions, etc.). Being probabilistic, the inference of neural networks can be measured by their sensitivity and specificity just as other more sequence-based motifs are.

3 Descriptions of Several Useful Motif Databases

The purpose of the following section on example databases of protein functional motifs is not to be an extensive review, but rather to give a few critical examples of useful resources for protein functional inference based on motifs. The web locations of the motif databases and methods is given in Table 1; however, one can readily find each by a simple Google search.

3.1 The Prosite Database

The Prosite database was one of the first family-based databases of protein consensus sequences. It was originally built largely by hand from well-characterized protein families that shared a common structural or functional

Table 1 Web locations of resources mentioned in this chapter

3MATRIX	http://3matrix.stanford.edu
3MOTIF	http://3motif.stanford.edu
BLAST search	http://www.ncbi.nlm.nih.gov/BLAST
Blocks database	http://blocks.fhcrc.org
CATH	http://www.biochem.ucl.ac.uk/bsm/cath
CBS prediction servers	http://www.cbs.dtu.dk/services
CBS Tools	http://www.cbs.dtu.dk/biotools
ClustalW alignment	http://www.ebi.ac.uk/clustalw
CODEHOP	http://blocks.fhcrc.org/blocks/codehop.html
DIALIGNment	http://dialign.gobics.de
eBLOCKs	http://eblocks.stanford.edu
eMATRIX	http://ematrix.stanford.edu
eMOTIF	http://emotif.stanford.edu
ePROTEOME	http://eproteome.stanford.edu
Genes3D	http://www.biochem.ucl.ac.uk/bsm/cath/Gene3D
HMMer home	http://hmmer.wustl.edu
InterPro	http://www.ebi.ac.uk/interpro
iProClass	http://pir.georgetown.edu/iproclass
ModBase	http://salilab.org/modbase
Molecular structure database	http://www.ebi.ac.uk/msd/index.html
MultiCoil	http://multicoil.lcs.mit.edu/cgi-bin/multicoil
PairCoil	http://paircoil.lcs.mit.edu/cgi-bin/paircoil
Panther	https://panther.appliedbiosystems.com
Pfam database	http://www.sanger.ac.uk/Software/Pfam
Phylogenomics	http://phylogenomics.berkeley.edu
PrePrints database	http://umber.sbs.man.ac.uk/dbbrowser/prePRINTS
PRINTS database	http://umber.sbs.man.ac.uk/dbbrowser/PRINTS
Prints-S database	http://umber.sbs.man.ac.uk/dbbrowser/sprint
ProDom	http://protein.toulouse.inra.fr/prodom.html
Prosite database	http://www.expasy.org/prosite
Protein Data Bank (PDB)	http://www.rcsb.org/pdb
SAM home	http://www.soe.ucsc.edu/research/compbio/sam.html
SCOP	http://scop.mrc-lmb.cam.ac.uk/scop
SMART	http://smart.embl-heidelberg.de
SuperFamily	http://supfam.mrc-lmb.cam.ac.uk/SUPERFAMILY
SWISS-MODEL	http://swissmodel.expasy.org/SWISS-MODEL.html
Swiss-Prot database	http://www.expasy.ch/sprot/sprot-top.html
T-COFFEE alignment	http://www.ch.embnet.org/software/TCoffee.html
TIGRFams	http://www.tigr.org/TIGRFAMs
UniProt	http://www.ebi.ac.uk/uniprot

site [69]. Protein sequences in the family were aligned using both algorithmic and manual methods. The conserved patterns were optimized to maximize their sensitivity on the known family and to minimize the false inferences. The manual curation of the motifs invariably leads to overfitting of the data. As mentioned above, the Prosite database gives measures of the sensitivity and specificity of the motifs derived from the current Swiss-Prot database, but these measures are overestimates of the expected performance on future sequence data. A true cross-validation approach to estimating the sensitivity and specificity of the Prosite motifs would be very tedious due to the manual motif generation. Another possible approach would be to examine motifs constructed several years ago on proteins characterized subsequently using nonsequence based approaches.

The Prosite database is distinguished by its excellent manual curation and by its excellent annotations. The Prosite and Swiss-Prot databases were among the first to have extensive links to other molecular biology databases, even before the appearance of the Internet.

The Prosite database contains a set of very common motifs for sites of post-translational modification. Since these motifs are very short and not of high information content, they may only be used to discover hypothetical modification sites which must be validated experimentally. Other, more sophisticated models of post-translational modification sites involving network and HMM models given higher confidence inferences [31,70].

Even many of the Prosite patterns which are not post-translational modifications have a relatively low specificity, on the order of 0.1–1% false predictions. Due to this low specificity, these patterns are most useful when scanning a single protein to infer functional sites. The Prosite pattern database cannot be used to infer function on entire proteomes, as they will result in more false inferences than true ones.

The Prosite database has also developed quantitative models for motifs called Prosite Profiles [17,36]. These Profiles are an extension of the original profile of Gribskov that characterized protein families [54, 55, 57]. Like those profiles, the Prosite Profiles estimate the likelihood of insertion and deletion at each position in the motif. Prosite Profiles are intermediate models between PSSMs and HMMs in that they have both position specific insertion and deletion penalties, but they do not have the full range of transition probabilities used by the Haussler or Baldi groups. Since the Prosite Profiles are quantitative, one can use different thresholds to gain sensitivity or specificity depending on the protein or proteome to be searched.

Recently, the Prosite group had developed a ProRule database in which functional and structural information is mapped onto the Prosite Profiles [103]. ProRules are short sequence patterns that identify the most critical residues for structure or function in the profile. These patterns are often less specific than Prosite patterns themselves and they are designed to be used to annotate a positive hit from the Prosite Profile. As such, the ProRule points out the critical catalytic or functional residue in the site. For example, the ProRules for the three trypsin family Profiles annotate the active site histidine, aspartate and serine residues of the catalytic triad (Figure 1). Combining Prosite Profiles with Prosite-type patterns gives ProRules the power of both quantitative and qualitative motifs simultaneously. ProRules also permits a much greater amount of biological information to be encoded in the motif.

3.2 The Blocks Databases

The Blocks and PRINTS databases are among the first PSSM representations of conserved protein motifs [10, 11, 62, 64]. Scoring matrices are probabilistic representations of conserved protein regions, usually derived from ungapped, multiply aligned protein subsequences. Posfai et al. originally applied the term *block* to a conserved, ungapped protein alignments for conserved, functional regions they discovered in families of restriction and modification enzymes [100, 101]. Multiple sequence alignments of type II restriction and modification enzymes from many bacteria strains showed conserved regions involved in cofactor (SAM) binding and catalytic sites. The DNA-binding sites were highly variable both in sequence and length – as was expected since the sequence specificity varied from strain to strain. Posfai et al. discovered that many of these conserved regions contained three or four positions (columns) that were completely conserved suggesting that these residues were especially important for function.

Smith and coworkers used this observation for discovering conserved blocks in other protein families [108]. They developed an algorithm called MOTIF that discovered short conserved sequence patterns containing three different positions separated by zero to eight nonconserved residues in a contiguous amino-acid region in each member of a protein family. Regions containing these short three position motifs were aligned and adjacent regions were examined for conservation as well. Using a heuristic method, they built up blocks of conserved residues centered on the original three position motifs.

Henikoff and Henikoff then automated this procedure so that they could be applied to any protein family and built up blocks around conserved functional sites [64, 121]. The initial Blocks database was limited to the same protein families found in the Prosite database. Later, the automatic Blocks approach was extended to other protein families including PRINTS, Pfam, DOMO and Prodom families. This extended database was known as the BLOCKS+ database [60, 65]. In all cases, the Blocks from these families were built using the automated method that found a conserved triplet of amino acids each separated by up to eight residues. This approach limits the diversity of conserved regions that Blocks can discover. Blocks, like its predecessor MOTIF, requires a large fraction (usually 90% or more) of the sequences to contain a perfectly conserved triplet of amino acids within a segment three to 19 amino acids in length. The automated Block discovery program cannot discover blocks that do not have a perfectly conserved three-amino-acid motif.

The function of many of these conserved regions is known through studies of the protein family from which they derive and from other functional studies.

The Blocks website provides several useful functionalities. The most useful for inference is Blocks Searcher that compares the blocks database against a protein sequence or a translated DNA sequence. Another function, Block-Maker and Multiple Alignment Processor, can take a multiple sequence alignment and generate a series of conserved ungapped alignment blocks. Finally, CodeHop can develop the best DNA probe to detect the DNA encoding the Block function taking into account the sequence variability in the Block and the codon usage table for the amino acids [102].

The Block Searcher function identifies all regions similar to each Block in the database and gives the expectation value for each hit. If there are multiple blocks discovered from the same family, a function called Block-scan will then calculate the expectation value for finding all the blocks simultaneously. Finally, Block Searcher prints out the relative position of the blocks in the query sequence and compares them with the relative positions of the same blocks in the Block Family database. These statistics are very useful for identifying the function novel proteins and they overcome the major deficiency that multiple motifs are usually required to encode a single biological function.

3.3 The PRINTS Database

The PRINTs database was initially constructed in a manual approach similar to the Prosite database. Initially it began with a database of conserved regions within signal transduction proteins (G-proteins in particular); however, it was extended to most other families [9, 10, 13]. An automated method for taking protein alignments from ProDom and converting them to ungapped conserved regions and automatically annotating them generates a database referred to as PrePrints database [11, 35, 42]. Once these regions are curated and verified, they pass into the PRINTs database. A newer relational version of the PRINTS database referred to as Prints-S is also available for relational searches [12]. The ProDom database itself is built up by PSI-BLAST alignments of protein families which results in a database of conserved protein domains.

One of the useful features of the PRINTs database is that it tabulates how many members of the protein family contain all the conserved regions known in that family, say N motifs), how many proteins contain $N-1$ motifs, $N-2$ motifs, etc. Thus one can see which motifs are present in all members of a protein family and which motifs may add optional functionality.

3.4 The eBLOCKs Database

The eBLOCKs database contains ungapped sequence alignments of all conserved regions within the Swiss-Prot protein database [114]. These conserved regions include functional sites such as catalytic sites, substrate binding site, cofactor binding sites and protein–protein interaction sites. The eBLOCKs database is built by running all Swiss-Prot proteins as queries against all other Swiss-Prot proteins using PSI-BLAST, collecting all family alignments and then extracting all ungapped conserved regions within each family. Unlike the Blocks and Motif methods, eBLOCKs does not require three highly conserved positions in the conserved regions. The minimum requirements for a block in the eBLOCK database are that the block must be at least 10 amino acids wide, three sequences deep and contain at least 40 bits of information. These criteria eliminate common motifs such as post-translation modification motifs, and ensure that the resulting BLOCKs have enough specificity to be used for searching entire proteomes.

The eBLOCKs database also contains multiple overlapping blocks for many conserved regions. This permits having very sensitive motifs that can cover a wide taxonomic diversity of the functional region, as well as more specific motifs that cover the same functional site in a subset of species (bacterial, eukaryotic, vertebrate, primate, etc.). The more general eBLOCKs increase the sensitivity of motifs inference, while the more specific eBLOCKs can identify not only the function, but often the taxonomic family containing the motif. This also results in a redundancy in the eBLOCK output. This redundancy is minimized by presenting the most specific hit first and then only showing additional, more general motifs if the user chooses to drill down in the results page.

About 35% of conserved regions in eBLOCKs are similar to functionally conserved regions in the Prosite, Blocks, PRINTS and InterPro databases. About 65% are novel conserved regions whose precise molecular function is not known, but which may represent protein–protein interaction sites, protein–ligand interaction sites or other functional sites not yet characterized in known protein families. Even though the precise molecular function of these conserved regions is not known, one does know the family of Swiss-Prot proteins from which the conserved region was derived, thus permitting the inference of family membership, if not specific molecular function. Unlike other protein family and domain databases (ProDom, Pfam, TIGRFams, etc. [21, 35, 42, 58]), eBLOCKs only records the ungapped conserved sites. However, like Blocks [60] and PRINTS [11], eBLOCKs keeps all the conserved regions from one family together in a group. The eBLOCK-conserved regions are converted into both sequence patterns (eMOTIFs) [68, 91] and into PSSMs (eMATRICES) [126, 127].

3.5 The eMOTIF Database

The eMOTIF algorithm for building sequence patterns from alignments is completely automated, and results in multiple motifs for different subsets of the sequences and also multiple motifs at different levels of specificity [68,91]. Rather than attempting to make a single motif to cover an entire protein family, the eMOTIF build algorithm attempts to find a well-conserved subset of the family where it can construct a highly specific motif that will infer just that subset. In order to maintain sensitivity, eMOTIF then attempts to find motifs for additional highly conserved subsets. By using a disjunction of several motifs, each of which covers a different subset of sequences, eMOTIF can maintain a high degree of sensitivity and a very high specificity. The eMOTIF build process effectively subclassifies the protein family and builds motifs that are specific enough to scan entire proteomes with few false inferences. The eMOTIF build process generates multiple motifs that span the entire space of sensitivity and specificity.

In order to minimize overfitting of the training data, eMOTIF is prevented from using arbitrary subsets of amino acids at any position in the motif. Instead, eMOTIF is limited to using only individual amino acids or 20 specified groups of amino acids that have been determined to the 20 most significant sets of amino acids found in protein multiple sequence alignments [126]. These groups of amino acids have been determined by examining the statistical significance of all possible subsets of the 20 amino acids in protein alignments. Only 20 subsets were significantly overrepresented. These subsets of amino acids include hydrophobic, hydrophilic, charged, basic, acidic, small, β-branched, hydrogen bond donor and acceptor classes among others. By limiting the groups of amino acids in motifs to these 20 subsets and the 20 individual amino acids, we prevent the motif from being influenced by outliers and from overfitting the training set data.

The eMOTIF database was originally built from ungapped conserved alignments from Blocks, PRINTS and InterPro protein families [68]. More recently it is built from the eBLOCKs database, but with links from similar families in other databases [114]. The eMOTIF database can be used to scan either individual proteins or entire proteomes. For scanning entire proteomes, one must choose a set of eMOTIFs with a specificity that will give a low false discovery rate. eMOTIF scans of all proteomes have been tabulated in the ePROTEOME database. ePROTEOME is an SQL database of all eMOTIFs found in all open reading frames in over 180 proteomes (Saxonov, Xu and Brutlag, unpublished; http://eproteome.stanford.edu).

eMOTIF has several advantages in searching for functional sites in proteomes. (i) One can chose sets of very high specificity motifs to minimize the false discovery rate, effectively eliminating false predictions. (ii) eMO-

TIF provides motifs targeted at many subsets of a protein family including functional subsets as well as taxonomic subsets. eMOTIFs, being discrete patterns, also indicate the most critical residues for function and the degree of residue variation permitted at each position. (iii) Since eMOTIFs are fixed-length regular expressions, eMOTIF search can be implemented in a number of very rapid search procedures including bitmap compare and trie algorithms that permit searching for thousands of motifs simultaneously. One can search an entire bacterial proteome for over 500 000 different motifs in less than an hour on a single processor machine. eMOTIFs have also been mapped onto all the structures in PDB so that one can readily see three dimensional examples of the eMOTIFs is the 3MOTIF database [26].

3.6 The eMATRIX Database

The original eMATRIX database was a database of PSSMs build from protein alignments in Blocks, PRINTS and InterPro [126, 127]. More recently it includes scoring matrices built from the eBLOCKs database [114]. The primary differences between the eMATRIX approach and other PSSM methods are sensitivity and speed. As mentioned before, eMATRIX calculates its scores using a variable number of pseudocounts for each matrix. The calculation of the correct number of pseudocounts is based on the minimal risk criterion [127]. This approach can increase the sensitivity of eMATRICES over other PSSMs by as much as 25%.

In addition, the inference procedure used by eMATRIX is 100–500 times faster than other methods for two reasons [126]. Prior to performing an eMATRIX scan of a protein or a proteome, eMATRIX takes the minimal threshold probability and converts it to the minimal score that will meet this threshold. At each point in the scan of a protein segment, eMATRIX can tell if it can meet this threshold score. Most often, the initial score for the beginning of a comparison will be so negative that eMATRIX need not finish scoring the entire segment since it knows it can never exceed the threshold score. In this way, eMATRIX only calculates the score for protein segments that do, in fact, meet the threshold. Most protein segments (above 99.9% for a threshold probability of 0.001) never have their score calculated speeding up the inference procedure markedly. The second speed-up comes because eMATRIX does not calculate the segment score in the linear order of the sequence. eMATRIX sorts the columns of the PSSM in order of information content with the most highly conserved positions first and the least highly conserved positions last. By scoring segments in order of conservation; one can tell immediately if a particular protein segment is going to match the matrix after scoring only one or two positions. Taken together, using the minimum threshold for stopping segment scoring early and sorting the scoring matrix by conservation gives a

100- to 500-fold speed-up depending on the specificity of the matrix and its overall length. This increased speed is very important both for interactive use and for the ability to rebuild functional assignment databases as the eMATRIX database is incrementally improved. The eMATRIX database can scan an entire bacterial proteome for 156 000 functional motifs in less time than most other PSSM databases can scan a single protein sequence. Using this method ePROTEOME has been developed which contains all eMATRIX hits in over 180 proteomes.

Because of its more powerful representation, eMATRIX, like other PSSM databases (Blocks, PRINTS, Prosite Profile) generally shows a higher sensitivity at discovering protein functional sites than do eMOTIF or the Prosite Pattern databases. Nevertheless, the use of sequence patterns such as eMOTIFs, Prosite Patterns and ProRules are very useful for identifying the most critical residues for function and which residues are permitted at those positions. All of the eMATRIX motifs have also been mapped onto PDB structures so that examples of the three dimensional structure of the motifs may be observed [25].

3.7 HMM Databases

Unlike the sequence pattern and PSSMs, the HMMs of protein families are more global, representing longer regions and including gapped alignments. HMMs are often used to represent protein families (Pfam [21], TIGRFams [58]) as well as protein structural domains [3, 35, 53, 97]. Since the function of a protein is often linked to family membership, if one can find the family membership of new protein sequence, it will aid tremendously in helping to determine the function of the novel protein. Many of these HMM-based databases are built specifically to aide in proteome annotation. As mentioned above, HMM models have the advantage of linking multiple ungapped motifs together in a probabilistic framework that preserves the order of the conserved functional and structural regions, and can discount the less-conserved regions between them. The ability of an HMM to infer the functional class of a novel protein usually requires that some members of the training set be taxonomically close to the target sequence. For instance, if one were to use HMM models for protein families made from bacterial sequences alone, the models would not be as sensitive when attempting to infer the function of eukaryotic proteins. The same can be said of PSSM and sequence motifs. Ideally, one would like to have multiple HMM models that could span both the functional and the taxonomic axes. There are significant efforts along these lines [105, 116, 117].

HMM models can be used to model short motif regions as well as entire families. By using transition states that represent different regions of a protein, rather than different residues, models of signal peptides, transmembrane helices and even entire transmembrane proteins have been possible [71, 80, 111]. The TMHMM model for transmembrane proteins has different states for the membrane, cytoplasmic and extracellular segments of the protein.

3.8 The InterPro Database

The InterPro Database [4, 90] is a large integrated protein functional annotation resource. The InterPro database integrates several motif-finding algorithms including Prosite [69], PRINTS [11], SMART [81], ProDom [35], Pfam [21], TIGRFams [58], iProClass [67] and Panther [88, 116]. InterPro also contains several structural databases including SCOP [3], CATH [97], Superfamily [53], Gene3D [97] as well as protein structure modeling applications including SWISS-MODEL [76] and ModBase [99]. By integrating these resources into one site, InterPro permits a user to perform many of the relevant analyses on a single protein sequence in one step. The results are both tabulated and presented graphically in linear formats that show the location of the motifs, domains, family and other functional sites along the protein sequence. The InterPro interface basically takes the query and applies each inference engine to it independently and then integrates the output into a single graphical and tabular result. This approach is very attractive because it shows the relationship between the results of many methods. On the other hand, it gives the illusion that many of the functional motif-finding methods are in agreement with each other. The problem is that these methods are not independent from each other; often they are based on different motifs made with the same starting families. InterPro publications state that due to its internal consistency checking it can provide deeper coverage (higher sensitivity) for detection of functional sites. The current InterPro database covers 77% of the UniProt [6] protein database.

Many of the methods included in InterPro have associated measures of specificity, expectation or false discovery rates; however, the user has no ability to change the thresholds for defining a valid hit. Also several methods do not have reliable measures of expectation and hence the results must be examined carefully. Fortunately, the curators have assigned a match status to each hit of InterPro on the protein sequence database including a match descriptor of T for a known true hit, F for a known false positive, N for a known false negative, P for a partial hit and "?" for a match of unknown status. These match status markers can be valuable adjunct to ones confidence in the inferred site.

3.9 Supervised versus Unsupervised Learning of Motifs

Many motif databases have been specifically derived from known protein families with well-defined functional sites and often structures. This is the case for the Prosite patterns, Blocks, PRINTS, Pfam, TIGRFams and SMART motif databases. Many structural domain databases are also built from proteins of known fold, family or subfamily (SuperFamily and Gene3D are two examples). This procedure of building motifs from a well-defined subset of proteins is a classic example of supervised learning. Protein sequences are separated into those that are known to have the requisite function or structure from those that do not. The goal is to develop motifs that can discriminate between the two classes.

An alternative approach is to build motifs in an unsupervised way by comparing all protein sequences to all others, discovering protein families and conserved regions within these families driven only by sequence similarity. This data-driven approach has the advantage that it can find families of proteins and functional motifs for functions that have not yet been discovered or characterized experimentally. This is particularly important for learning protein–protein interaction motifs and other functions that have not been well explored experimentally. Of course unsupervised learning has the disadvantage that many of the conserved regions are anonymous, with their precise functions unknown. One of course learns the family of proteins containing the conserved motif, but not the specific function of the motif for that family.

One approach to annotating anonymous motifs is to associate the motifs with certain descriptors present in the protein sequence annotations. These descriptors can be functional such as Enzyme Commission (EC) classification numbers [16,41], Gene Ontology (GO) terms [7,82], substrate or cofactor binding sites [113], etc., or they could be more general classes such as taxonomic terms, protein family names or properties, or even uncontrolled vocabulary terms such as keywords. Some databases, such as InterPro, have assigned GO terms and other functional descriptors to motifs just based on the GO terms associated with the sequences identified by the motif.

A better approach is to classify the motifs themselves using standard classification methods. For example, using support vector machine (kernel methods) or *k*-nearest-neighbor classification one can discover which motifs from a large collection of anonymous motifs are predictive of enzyme mechanism (EC classification number) [22, 23]. EC numbers are determined experimentally and are assigned manually to each of the enzymes in the Swiss-Prot database; hence, they form a well-defined functional description that can serve as a "gold standard" for enzyme function. Classifying both individual and groups of motifs based on their ability to predict enzyme classification number overcomes the problem that biological functions may be encoded by multiple motifs. Using groups of motifs can also increase the specificity and sensitivity of motifs for predicting biological function. This classification of motifs by function also facilitates the annotation of anonymous motifs.

Another example of learning the function of motifs by comparison with other databases comes from the work of Wang and coworkers [122]. By examining motifs contained in proteins in an extensive protein–protein interaction network, they were able to learn pairs of motifs that were the most likely to be involved in the protein–protein interaction. The motif pairs were able to predict the interactions in nearly 70% of the highly significant interactions in yeast as well as 70% of the yeast protein structures in the Protein Structure Database. Similar efforts to predict interacting domains represented by HMMs have also been successful [43,112].

4 Summary and Conclusions

In this chapter, we have attempted to describe the primary methods for representing conserved functional regions, motifs, in protein sequences. Each method has its own advantages and limitations for representing function and for inferring function in novel proteomes. One of the most important considerations when attempting to infer the function of all the proteins in a proteome is the false discovery rate. As there are so many genes and so many functional motifs, even a relatively specific motif can yield more false inferences than correct ones when applied to an entire proteome. This is particularly true if one is using pairwise sequence similarities to draw functional inferences. Motifs discussed in this chapter have the advantage of providing a family-wise comparison that focuses on the most highly conserved regions and residues in the functional site. By focusing on the conserved residues one increases the signal-to-noise ratio tremendously and permits much more accurate functional inference.

In addition to sequence-based motifs, we have also mentioned network-based motifs and other quantitative motifs that can represent patterns of size, charge, hydrophobicity and other attributes not necessarily obvious in the sequences themselves. These network-based motifs are often more flexible at representing biological function and structural features than sequence based motifs. They also have the ability to represent correlated changes in protein sequence and structure that often elucidate features that nature is conserving that are sometimes hidden by pure sequence representations.

We have discussed two main approaches to learning motifs. The most common is to discover conserved regions in small sets of proteins known to have a common structure or function. This is a supervised method for learning motifs in which biological knowledge is provided via the training set of proteins. The unsupervised approach is to run a systematic comparison of all proteins against all others without regard to function. In such an unsupervised approach one learns the protein families, subfamilies and conserved regions or blocks. Functional annotation can be assigned to these conserved regions either through their component sequences or via motif classification approaches. Unfortunately, due to space limitations, we have not been able to discuss all the different methods for learning the different types of motifs in depth.

Finally, motifs are a compact representation of the critical structural and functional components of proteins. Rather than having to compare new protein sequences against all other protein sequences which are growing exponentially, comparison against a much smaller set of well known conserved regions from protein families is more accurate and much more rapid method for inferring function.

References

1 ALTSCHUL, S. F., W. GISH, W. MILLER, E. W. MYERS AND D. J. LIPMAN. 1990. Basic local alignment search tool. J. Mol. Biol. **215**: 403–10.

2 ALTSCHUL, S. F., T. L. MADDEN, A. A. SCHAFFER, J. ZHANG, Z. ZHANG, W. MILLER AND D. J. LIPMAN. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**: 3389–402.

3 ANDREEVA, A., D. HOWORTH, S. E. BRENNER, T. J. HUBBARD, C. CHOTHIA AND A. G. MURZIN. 2004. SCOP database in 2004: refinements integrate structure and sequence family data. Nucleic Acids Res. **32**: D226–9.

4 APWEILER, R., T. K. ATTWOOD, A. BAIROCH, et al. 2000. InterPro – an integrated documentation resource for protein families, domains and functional sites. Bioinformatics **16**: 1145–50.

5 APWEILER, R., T. K. ATTWOOD, A. BAIROCH, et al. 2001. The InterPro database, an integrated documentation resource for protein families, domains and functional sites. Nucleic Acids Res. **29**: 37–40.

6 APWEILER, R., A. BAIROCH, C. H. WU, et al. 2004. UniProt: the Universal Protein knowledgebase. Nucleic Acids Res. **32**: D115–9.

7 ASHBURNER, M., C. A. BALL, J. A. BLAKE, et al. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. **25**: 25–9.

8 ATTESON, K. 1998. Calculating the exact probability of language-like patterns in biomolecular sequences. Proc. ISMB **6**: 17–24.

9 ATTWOOD, T. K. AND M. E. BECK. 1994. PRINTS – a protein motif fingerprint database. Protein Eng. **7**: 841–8.

10 ATTWOOD, T. K., M. E. BECK, A. J. BLEASBY AND D. J. PARRY-SMITH. 1994. PRINTS – a database of protein motif fingerprints. Nucleic Acids Res. **22**: 3590–6.

11 ATTWOOD, T. K., P. BRADLEY, D. R. FLOWER, et al. 2003. PRINTS and its automatic supplement, prePRINTS. Nucleic Acids Res. **31**: 400–2.

12 ATTWOOD, T. K., M. D. R. CRONING, D. R. FLOWER, A. P. LEWIS, J. E. MABEY, P. SCORDIS, J. N. SELLEY AND W. WRIGHT. 2000. PRINTS-S: the database formerly known as PRINTS. Nucleic Acids Res. **28**: 225–7.

13 Attwood, T. K. and J. B. Findlay. 1993. Design of a discriminating fingerprint for G-protein-coupled receptors. Protein Eng. **6**: 167–76.

14 Babbitt, P. C. 2003. Definitions of enzyme function for the structural genomics era. Curr. Opin. Chem. Biol. **7**: 230–7.

15 Bairoch, A. 1991. Prosite: a dictionary of sites and patterns in protein (release 6.1). Nucleic Acids Res. **19**: 2241–5.

16 Bairoch, A. 2000. The ENZYME database in 2000. Nucleic Acids Res. **28**: 304–5.

17 Bairoch, A. and P. Bucher. 1994. PROSITE: recent developments. Nucleic Acids Res. **22**: 3583–9.

18 Baldi, P., T. Chauvin, M. Hunkapiller and A. McClure. 1992. Hiden Markov Models in Molecular Biology: New Algorithms and Applications. In Proc. Neural Information Processing Systems, Denver, CO: 747–54.

19 Baldi, P. and Y. Chauvin. 1995. Protein modeling with hybrid Hidden Markov Model/neural network architectures. Proc. ISMB **3**: 39–47.

20 Baldi, P., Y. Chauvin, T. Hunkapiller and M. A. McClure. 1994. Hidden Markov models of biological primary sequence information. Proc. Natl Acad. Sci. USA **91**: 1059–63.

21 Bateman, A., L. Coin, R. Durbin, et al. 2004. The Pfam protein families database. Nucleic Acids Res. **32**: D138–41.

22 Ben-hur, A. and D. Brutlag. 2003. Remote homology detection: a motif based approach. Bioinformatics **19**: i26–33.

23 Ben-Hur, A. and D. L. Brutlag. 2005. Protein sequence motifs: Highly predictive features of protein function. In Guyon, I., S. Gunn, M. Nikravesh and L. Zadeh (eds.), *Feature Extraction, Foundations and Applications*. Springer, Berlin, Chapter 31: 625–45.

24 Bendtsen, J. D., H. Nielsen, G. von Heijne and S. Brunak. 2004. Improved prediction of signal peptides: SignalP 3.0. J. Mol. Biol. **340**: 783–95.

25 Bennett, S. P., L. Lu and D. L. Brutlag. 2003. 3MATRIX and 3MOTIF: a protein structure visualization system for conserved sequence motifs. Nucleic Acids Res. **31**: 3328–32.

26 Bennett, S. P., C. G. Nevill-Manning and D. L. Brutlag. 2003. 3MOTIF: visualizing conserved protein sequence motifs in the protein structure database. Bioinformatics **19**: 541–2.

27 Berger, B. 1995. Algorithms for protein structural motif recognition. J. Comput. Biol. **2**: 125–38.

28 Berger, B. and M. Singh. 1997. An iterative method for improved protein structural motif recognition. J. Comput. Biol. **4**: 261–73.

29 Berger, B., D. B. Wilson, E. Wolf, T. Tonchev, M. Milla and P. S. Kim. 1995. Predicting coiled coils by use of pairwise residue correlations. Proc. Natl Acad. Sci. USA **92**: 8259–63.

30 Blom, N., S. Gammeltoft and S. Brunak. 1999. Sequence and structure-based prediction of eukaryotic protein phosphorylation sites. J. Mol. Biol. **294**: 1351–62.

31 Blom, N., T. Sicheritz-Ponten, R. Gupta, S. Gammeltoft and S. Brunak. 2004. Prediction of post-translational glycosylation and phosphorylation of proteins from the amino acid sequence. Proteomics **4**: 1633–49.

32 Boeckmann, B., A. Bairoch, R. Apweiler, et al. 2003. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. Nucleic Acids Res. **31**: 365–70.

33 Bradley, P., P. S. Kim and B. Berger. 2002. TRILOGY: Discovery of sequence-structure patterns across diverse proteins. Proc. Natl Acad. Sci. USA **99**: 8500–5.

34 Brown, M., R. Hughey, A. Krogh, I. S. Mian, K. Sjolander and D. Haussler. 1993. Using Dirichlet mixture priors to derive hidden Markov models for protein families. Proc. ISMB **1**: 47–55.

35 Bru, C., E. Courcelle, S. Carrere, Y. Beausse, S. Dalmar and D. Kahn. 2005. The ProDom database of protein

domain families: more emphasis on 3D. Nucleic Acids Res. **33**: D212–5.

36 BUCHER, P. AND A. BAIROCH. 1994. A generalized profile syntax for biomolecular sequence motifs and its function in automatic sequence interpretation. Proc. ISMB **2**: 53–61.

37 CARTER, P. AND J. A. WELLS. 1988. Dissecting the catalytic triad of a serine protease. Nature **332**: 564–8.

38 CHOTHIA, C. AND A. M. LESK. 1987. The evolution of protein structures. Cold Spring Harb. Symp. Quant. Biol. **52**: 399–405.

39 CHOTHIA, C. AND A. M. LESK. 1986. The relation between the divergence of sequence and structure in proteins. EMBO J. **5**: 823–6.

40 CLAROS, M. G., S. BRUNAK AND G. VON HEIJNE. 1997. Prediction of N-terminal protein sorting signals. Curr. Opin. Struct. Biol. **7**: 394–8.

41 COMMISSION, I.-I. 1972. IUPAC–IUB Commission on Biochemical Nomenclature (CBN). Nomenclature of multiple enzyme types. Recommendations 1971. Hoppe Seylers Z. Physiol. Chem. **353**: 852–4.

42 CORPET, F., F. SERVANT, J. GOUZY AND D. KAHN. 2000. ProDom and ProDom-CG: tools for protein domain analysis and whole genome comparisons. Nucleic Acids Res. **28**: 267–9.

43 DENG, M., S. MEHTA, F. SUN AND T. CHEN. 2002. Inferring domain–domain interactions from protein–protein interactions. Genome Res. **12**: 1540–8.

44 DEVOS, D. AND A. VALENCIA. 2001. Intrinsic errors in genome annotation. Trends Genet. **17**: 429–31.

45 DEVOS, D. AND A. VALENCIA. 2000. Practical limits of function prediction. Proteins **41**: 98–107.

46 DOIG, A. J., A. CHAKRABARTTY, T. M. KLINGLER AND R. L. BALDWIN. 1994. Determination of free energies of N-capping in alpha-helices by modification of the Lifson–Roig helix–coil therapy to include N- and C-capping. Biochemistry **33**: 3396–403.

47 EDDY, S. R. 1998. Profile hidden Markov models. Bioinformatics **14**: 755–63.

48 EDDY, S. R., G. MITCHISON AND R. DURBIN. 1995. Maximum discrimination hidden Markov models of sequence consensus. J. Comput. Biol. **2**: 9–23.

49 EISENBERG, D., E. SCHWARZ, M. KOMAROMY AND R. WALL. 1984. Analysis of membrane and surface protein sequences with the hydroponic moment plot. J. Mol. Biol. **179**: 125–142.

50 EISENBERG, D., R. M. WEISS AND T. C. TERWILLIGER. 1984. The hydrophobic moment detects periodicity in protein hydrophobicity. Proc. Natl. Acad. Sci. **81**: 140–144.

51 EMANUELSSON, O., H. NIELSEN, S. BRUNAK AND G. VON HEIJNE. 2000. Predicting subcellular localization of proteins based on their N-terminal amino acid sequence. J. Mol. Biol. **300**: 1005–16.

52 GERLT, J. A. AND P. C. BABBITT. 2000. Can sequence determine function? Genome Biol. **1**: REVIEWS0005.

53 GOUGH, J. AND C. CHOTHIA. 2002. SUPERFAMILY: HMMs representing all proteins of known structure. SCOP sequence searches, alignments and genome assignments. Nucleic Acids Res. **30**: 268–72.

54 GRIBSKOV, M. 1994. Profile analysis. Methods Mol. Biol. **25**: 247–66.

55 GRIBSKOV, M., A. D. MCLACHLAN AND D. EISENBERG. 1987. Profile analysis: detection of distantly related proteins. Proc. Natl Acad. Sci. USA **84**: 4355–8.

56 GRIBSKOV, M. AND N. L. ROBINSON. 1996. use of receiver operating characteristic (ROC) analysis to evalutate sequence matching. Comput. Chem. **20**: 25–33.

57 GRIBSKOV, M. AND S. VERETNIK. 1996. Identification of sequence pattern with profile analysis. Methods Enzymol. **266**: 198–212.

58 HAFT, D. H., J. D. SELENGUT AND O. WHITE. 2003. The TIGRFAMs database of protein families. Nucleic Acids Res. **31**: 371–3.

59 HAUSSLER, D., A. KROGH, S. MIAN AND K. SJOLANDER. 1993. Protein

modeling using hidden Markov models: analysis of globins. Presented at the 26th Annu. Hawaii Int. Conf. on System Sciences: Architecture and Biotechnology Computing, Wailea, HI: 792–802.

60 Henikoff, J. G., E. A. Greene, S. Pietrokovski and S. Henikoff. 2000. Increased coverage of protein families with the blocks database servers. Nucleic Acids Res. **28**: 228–30.

61 Henikoff, J. G. and S. Henikoff. 1996. Using substitution probabilities to improve position-specific scoring matrices. Comput. Appl. Biosci. **12**: 135–43.

62 Henikoff, J. G., S. Pietrokovski, C. M. McCallum and S. Henikoff. 2000. Blocks-based methods for detecting protein homology. Electrophoresis **21**: 1700–6.

63 Henikoff, S. 1996. Scores for sequence searches and alignments. Curr. Opin. Struct Biol. **6**: 353–60.

64 Henikoff, S. and J. G. Henikoff. 1991. Automated assembly of protein blocks for database searching. Nucleic Acids Res. **19**: 6565–72.

65 Henikoff, S., J. G. Henikoff and S. Pietrokovski. 1999. Blocks+: a non-redundant database of protein alignment blocks derived from multiple compilations. Bioinformatics **15**: 471–9.

66 Hsu, F., T. H. Pringle, R. M. Kuhn, D. Karolchik, M. Diekhans, D. Haussler and W. J. Kent. 2005. The UCSC proteome browser. Nucleic Acids Res. **33**: D454–8.

67 Huang, H., W. C. Barker, Y. Chen and C. H. Wu. 2003. iProClass: an integrated database of protein family, function and structure information. Nucleic Acids Res. **31**: 390–2.

68 Huang, J. Y. and D. L. Brutlag. 2001. The EMOTIF database. Nucleic Acids Res. **29**: 202–4.

69 Hulo, N., C. J. Sigrist, V. Le Saux, et al. 2004. Recent improvements to the PROSITE database. Nucleic Acids Res. **32**: D134–7.

70 Jensen, L. J., R. Gupta, N. Blom, et al. 2002. Prediction of human protein function from post-translational modifications and localization features. J. Mol. Biol. **319**: 1257–65.

71 Juncker, A. S., H. Willenbrock, G. Von Heijne, S. Brunak, H. Nielsen and A. Krogh. 2003. Prediction of lipoprotein signal peptides in Gram-negative bacteria. Protein Sci. **12**: 1652–62.

72 Karplus, K. 1995. Evaluating regularizers for estimating distributions of amino acids. Proc. ISMB **3**: 188–96.

73 Kesmir, C., A. K. Nussbaum, H. Schild, V. Detours and S. Brunak. 2002. Prediction of proteasome cleavage motifs by neural networks. Protein Eng. **15**: 287–96.

74 Klingler, T. M. and D. L. Brutlag. 1994. Discovering side-chain correlation in alpha-helices. Proc. ISMB **2**: 236–43.

75 Klingler, T. M. and D. L. Brutlag. 1994. Discovering structural correlations in alpha-helices. Protein Sci. **3**: 1847–57.

76 Kopp, J. and T. Schwede. 2004. The SWISS-MODEL Repository of annotated three-dimensional protein structure homology models. Nucleic Acids Res. **32**: D230–4.

77 Krem, M. M. and E. Di Cera. 2001. Molecular markers of serine protease evolution. EMBO J. **20**: 3036–45.

78 Krem, M. M., T. Rose and E. Di Cera. 2000. Sequence determinants of function and evolution in serine proteases. Trends Cardiovasc. Med. **10**: 171–6.

79 Krogh, A., M. Brown, I. S. Mian, K. Sjolander and D. Haussler. 1994. Hidden Markov models in computational biology. Applications to protein modeling. J. Mol. Biol. **235**: 1501–31.

80 Krogh, A., B. Larsson, G. von Heijne and E. L. Sonnhammer. 2001. Predicting transmembrane protein topology with a hidden Markov model: application to complete genomes. J. Mol. Biol. **305**: 567–80.

81 Letunic, I., R. R. Copley, S. Schmidt, F. D. Ciccarelli, T. Doerks, J. Schultz, C. P. Ponting and P. Bork. 2004. SMART 4.0: towards genomic data integration. Nucleic Acids Res. **32**: D142–4.

82 LEWIS, S. E. 2005. Gene Ontology: looking backwards and forwards. Genome Biol. **6**: 103.

83 LIANG, M. P., D. R. BANATAO, T. E. KLEIN, D. L. BRUTLAG AND R. B. ALTMAN. 2003. WebFEATURE: An interactive web tool for identifying and visualizing functional sites on macromolecular structures. Nucleic Acids Res. **31**: 3324–7.

84 LIANG, M. P., D. L. BRUTLAG AND R. B. ALTMAN. 2003. Automated construction of structural motifs for predicting functional sites on protein structures. Pac. Symp. Biocomput. **8**: 204–15.

85 MADERA, M., C. VOGEL, S. K. KUMMERFELD, C. CHOTHIA AND J. GOUGH. 2004. The SUPERFAMILY database in 2004: additions and improvements. Nucleic Acids Res. **32**: D235–9.

86 WU, S. AND U. MANBER. 1992. Agrep – A fast approximate pattern-matching tool. Proceeding of USENIX Technical Conference, 153–62.

87 MCDONNELL, A. V., T. JIANG, A. E. KEATING AND B. BERGER. 2005. Paircoil2: improved prediction of coiled coils from sequence. Bioinformatics **22**: 356–8.

88 MI, H., B. LAZAREVA-ULITSKY, R. LOO, et al. 2005. The PANTHER database of protein families, subfamilies, functions and pathways. Nucleic Acids Res. **33**: D284–8.

89 MORGENSTERN, B., K. FRECH, A. DRESS AND T. WERNER. 1998. DIALIGN: finding local similarities by multiple sequence alignment. Bioinformatics **14**: 290–4.

90 MULDER, N. J., R. APWEILER, T. K. ATTWOOD, et al. 2005. InterPro, progress and status in 2005. Nucleic Acids Res. **33**: D201–5.

91 NEVILL-MANNING, C. G., T. D. WU AND D. L. BRUTLAG. 1998. Highly specific protein sequence motifs for genome analysis. Proc. Natl Acad. Sci. USA **95**: 5865–71.

92 NIELSEN, H., S. BRUNAK AND G. VON HEIJNE. 1999. Machine learning approaches for the prediction of signal peptides and other protein sorting signals. Protein Eng. **12**: 3–9.

93 NIELSEN, H., J. ENGELBRECHT, S. BRUNAK AND G. VON HEIJNE. 1997. A neural network method for identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. Int J. Neural. Syst. **8**: 581–99.

94 NIELSEN, H., J. ENGELBRECHT, S. BRUNAK AND G. VON HEIJNE. 1997. Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites. Protein Eng. **10**: 1–6.

95 NIELSEN, M., C. LUNDEGAARD, P. WORNING, S. L. LAUEMOLLER, K. LAMBERTH, S. BUUS, S. BRUNAK AND O. LUND. 2003. Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. Protein Sci. **12**: 1007–17.

96 NOTREDAME, C., D. G. HIGGINS AND J. HERINGA. 2000. T-Coffee: a novel method for fast and accurate multiple sequence alignment. J. Mol. Biol. **302**: 205–17.

97 PEARL, F., A. TODD, I. SILLITOE, et al. 2005. The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. Nucleic Acids Res. **33**: D247–51.

98 PETERSEN, S. B., H. BOHR, J. BOHR, S. BRUNAK, R. M. COTTERILL, H. FREDHOLM AND B. LAUTRUP. 1990. Training neural networks to analyse biological sequences. Trends Biotechnol. **8**: 304–8.

99 PIEPER, U., N. ESWAR, H. BRABERG, et al. 2004. MODBASE, a database of annotated comparative protein structure models, and associated resources. Nucleic Acids Res. **32**: D217–22.

100 POSFAI, J., A. S. BHAGWAT, G. POSFAI AND R. J. ROBERTS. 1989. Predictive motifs derived from cytosine methyltransferases. Nucleic Acids Res. **17**: 2421–35.

101 POSFAI, J., A. S. BHAGWAT AND R. J. ROBERTS. 1988. Sequence motifs specific for cytosine methyltransferases. Gene **74**: 261–5.

102 ROSE, T. M., E. R. SCHULTZ, J. G. HENIKOFF, S. PIETROKOVSKI, C. M. MCCALLUM AND S. HENIKOFF. 1998. Consensus-degenerate hybrid

oligonucleotide primers for amplification of distantly related sequences. Nucleic Acids Res. **26**: 1628–35.

103 Sigrist, C. J., E. De Castro, P. S. Langendijk-Genevaux, V. Le Saux, A. Bairoch and N. Hulo. 2005. ProRule: a new database containing functional and structural information on PROSITE profiles. Bioinformatics **21**: 4060–6.

104 Sing, T., O. Sander, N. Beerenwinkel and T. Lengauer. 2005. ROCR: visualizing classifier performance in R. Bioinformatics **21**: 3940–1.

105 Sjolander, K. 2004. Phylogenomic inference of protein molecular function: advances and challenges. Bioinformatics **20**: 170–9.

106 Sjolander, K., K. Karplus, M. Brown, R. Hughey, A. Krogh, I. S. Mian and D. Haussler. 1996. Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. Comput. Appl. Biosci. **12**: 327–45.

107 Skolnick, J. and J. S. Fetrow. 2000. From genes to protein structure and function: novel applications of computational approaches in the genomic era. Trends Biotechnol. **18**: 34–9.

108 Smith, H. O., T. M. Annau and S. Chandrasegaran. 1990. Finding sequence motifs in groups of functionally related proteins. Proc. Natl Acad. Sci. USA **87**: 826–30.

109 Sonnhammer, E. L., S. R. Eddy, E. Birney, A. Bateman and R. Durbin. 1998. Pfam: multiple sequence alignments and HMM-profiles of protein domains. Nucleic Acids Res. **26**: 320–2.

110 Sonnhammer, E. L., S. R. Eddy and R. Durbin. 1997. Pfam: a comprehensive database of protein domain families based on seed alignments. Proteins **28**: 405–20.

111 Sonnhammer, E. L., G. von Heijne and A. Krogh. 1998. A hidden Markov model for predicting transmembrane helices in protein sequences. Proc ISMB **6**: 175–82.

112 Sprinzak, E. and H. Margalit. 2001. Correlated sequence-signatures as markers of protein-protein interaction. J. Mol. Biol. **311**: 681–92.

113 Stuart, A. C., V. A. Ilyin and A. Sali. 2002. LigBase: a database of families of aligned ligand binding sites in known protein sequences and structures. Bioinformatics **18**: 200–1.

114 Su, Q. J., L. Lu, S. Saxonov and D. L. Brutlag. 2005. eBLOCKs: enumerating conserved protein blocks to achieve maximal sensitivity and specificity. Nucleic Acids Res. **33**: D178–82.

115 Swets, J. A. 1988. Measuring the accuracy of diagnostic systems. Science **270**: 1285–1293.

116 Thomas, P. D., M. J. Campbell, A. Kejariwal, et al. 2003. PANTHER: a library of protein families and subfamilies indexed by function. Genome Res. **13**: 2129–41.

117 Thomas, P. D., A. Kejariwal, M. J. Campbell, et al. 2003. PANTHER: a browsable database of gene products organized by biological function, using curated protein family and subfamily classification. Nucleic Acids Res. **31**: 334–41.

118 Thompson, J. D., D. G. Higgins and T. J. Gibson. 1994. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. **22**: 4673–80.

119 Tian, W. and J. Skolnick. 2003. How well is enzyme function conserved as a function of pairwise sequence identity? J. Mol. Biol. **333**: 863–82.

120 Walker, J. E., M. Saraste, M. J. Runswick and N. J. Gay. 1982. Distantly related sequences in the alpha- and beta-subunits of ATP synthase, myosin, kinases and other ATP-requiring enzymes and a common nucleotide binding fold. EMBO J. **1**: 945–51.

121 Wallace, J. C. and S. Henikoff. 1992. PATMAT: a searching and extraction program for sequence, pattern and block queries and databases. Comput. Appl. Biosci. **8**: 249–54.

122 Wang, H., E. Segal, A. Ben-hur, D. Koller and D. L. Brutlag. 2005. Identifying protein–protein interaction sites on a genome-wide scale. Adv. Neural Inf. Process. Syst. **17**: 1465–72.

123 WILSON, C. A., J. KREYCHMAN AND M. GERSTEIN. 2000. Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores. J. Mol. Biol. **297**: 233–49.

124 WOLF, E., P. S. KIM AND B. BERGER. 1997. MultiCoil: a program for predicting two- and three-stranded coiled coils. Protein Sci. **6**: 1179–89.

125 WU, T. D. AND D. L. BRUTLAG. 1996. Discovering empirically conserved amino acid substitution groups in databases of protein families. Proc. ISMB **4**: 230–40.

126 WU, T. D., C. G. NEVILL-MANNING AND D. L. BRUTLAG. 2000. Fast probabilistic analysis of sequence function using scoring matrices. Bioinformatics **16**: 233–44.

127 WU, T. D., C. G. NEVILL-MANNING AND D. L. BRUTLAG. 1999. Minimal-risk scoring matrices for sequence analysis. J. Comput. Biol. **6**: 219–35.

128 ZHANG, B., L. RYCHLEWSKI, K. PAWLOWSKI, J. S. FETROW, J. SKOLNICK AND A. GODZIK. 1999. From fold predictions to function predictions: automation of functional site conservation analysis for functional genome predictions. Protein Sci. **8**: 1104–15.

31
Analyzing Protein Interaction Networks

Johannes Goll and Peter Uetz

1 Introduction

Protein–protein interactions are essential for all biological processes [3]. Although protein interactions have been studied for decades, only recent advances have made them accessible to systematic computational analysis. (i) Large-scale experimental studies generate genomic and proteomic data at an ever-increasing rate, permitting us to analyze whole "interactomes" and related biological information. (ii) Recently established databases of protein–protein interactions make interaction data easily accessible to theoreticians and experimentalists alike. (iii) Increasing numbers of solved three-dimensional (3-D) structures of proteins and protein complexes enable us to study such assemblies in atomic detail *in silico*. However, in contrast to hundreds of completely sequenced genomes, only a handful of comprehensive interaction studies have been carried out and there is no organism for which all protein interactions are known. Thus, we are still at an early stage of computational interactome analysis [114]. The field has exploded since the first interaction maps were published in 1997 for a subset of yeast proteins and in 2000 for a genome-wide data set [40, 117]. In addition to the flood of data, there is an even bigger avalanche of computational studies that analyze interaction data sets (Figure 1). In this chapter we will describe methods for the computational analysis of interaction networks and some results from such studies. We will cover the validation of experimental data sets as well as the prediction of protein interactions (which is a related topic). We will describe graph-theoretic approaches as well as the integration of interaction data sets with other biological information. Finally, we will give a summary of our current knowledge on the evolution of protein interactions.

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

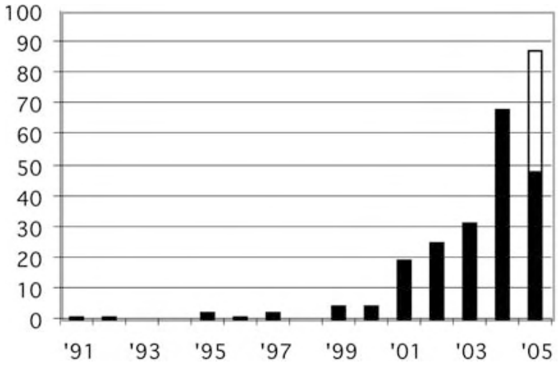


Figure 1 Publications on interactomes and protein networks. Number of references retrieved per year from PubMed by searching for *protein AND "interaction network" OR interactome* and the respective year (search performed on 18 July 2005).

2 Experimental Methods and Interaction Data

Before interaction data can be analyzed properly, their origin and nature need to be defined. In the past authors often compiled interaction data from different experiments into single data sets although the interactions were not really comparable. Thus, computational biologists need to know at least the basics about the experimental methods with which their data has been generated. For example, protein complexes are often treated as if all the interactions in them are known, although in most cases we do not know any of them (Figure 2, see also Refs. [35, 103] for further experimental details).

Although there are many experimental methods for analyzing protein interactions (Table 1), the bulk of the data has been produced with just a handful of them. The two most popular are the yeast *two-hybrid system* (Y2H [13, 34]) and *protein (or complex) copurification* in conjunction with tandem mass spectrometry (MS [88]). Their popularity mostly stems from the fact that both can be automated to a large extent and therefore carried out in a high-throughput fashion to produce large data sets of fairly consistent quality.

Y2H was the first to be used for several large-scale studies (reviewed in Ref. [133]). It uses two fusion proteins ("hybrids") whose interaction results in the activation of a reporter gene or enzyme whose activity can be detected or measured quantitatively [13, 14, 34].

The requirement for two hybrid proteins is one of the reasons why the system suffers from *false negatives*, i.e. physiological interactions that the method does not detect. The reason are most likely steric effects that prevent the proteins from interacting because of the fused domains. Another source of false negatives is the fact that these assays are usually carried out in yeast. If non-yeast proteins are tested they may not have post-translational mod-

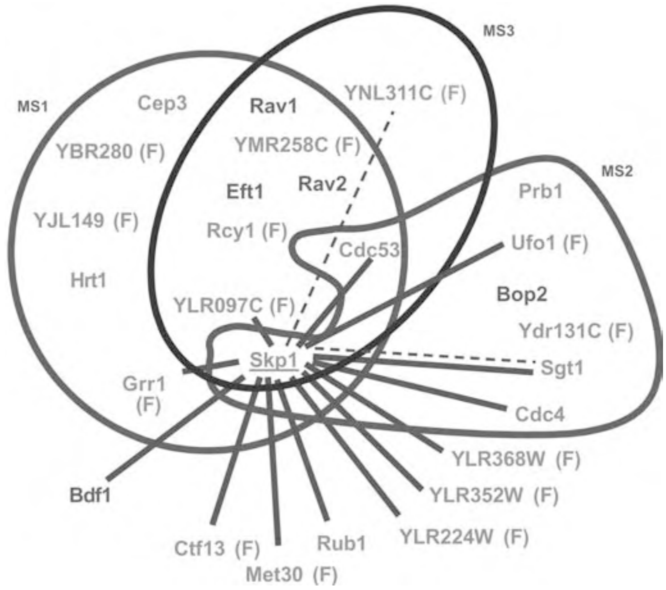


Figure 2 Interaction data gained by Y2H and MS. Skp1 is a protein involved in ubiquitin-mediated protein degradation, and has been epitope-tagged for both Y2H screens and MS analysis. The purified complexes of Skp1 from three independent MS studies (circles) and the binary interactions from two Y2H studies (solid and hatched lines) are shown. Despite the differences in the data sets, most of the discovered interactions seem to be plausible: most associated proteins are known to be involved in protein degradation. Skp1 is directed to its target proteins via so-called F-box proteins, which contain a short peptide motif, the F-box (F). (From Ref. [127].)

ifications that are necessary for interaction. Y2H has also a reputation of generating *false positives*, i.e. interaction signals that do not stem from real physical interactions. Most cases of false positives are not reproducible and are therefore not easy to explain. They are probably caused by mutations or other random events in the yeast cells used for the assay.

Protein *complex purification* (or copurification) and subsequent analysis by MS is the other major method for detecting protein interactions. However, this method is fundamentally different as it does not detect binary interactions (except in cases where only two proteins are copurified). Instead, a purified complex contains proteins that are held together by protein–protein interactions whose precise topology is usually not known (Figure 2).

Like Y2H, complex purification is also prone to generate false positives (proteins unspecifically binding to the complex) and false negatives (proteins that are lost from the native complex by too stringent washing).

Aloy and Russell [6] presented evidence that MS analysis of purified complexes tends to identify *stable complexes* while the Y2H assay tends to be biased towards *transient interactions*. Although it is difficult to classify experimentally

Table 1 Features of selected protein interaction detection methods

Class	Method	Analytical perspective[a]	Resolution level[a]	Advantages
Biochemical	affinity purification[b] + MS (MS)	perspective	complex	*in vivo*
	ELISA	analytical	protein	
	filter blot	analytical	protein	
	pull-down	analytical + perspective	protein/residues	
	comigration in non-denaturing gel	analytical	complex	*in vivo*
	Far Western[c]	analytical	protein	
	chemical cross-linking[d]	analytical + perspective	protein	
Biophysical	X-ray crystallography	analytical	atomic	
	NMR	analytical	atomic	
	surface plasmon resonance (Biacore)	analytical	protein	quantitative, kinetics
	isothermal titration calorimetry	analytical	protein	quantitative, kinetics
	scintillation proximity assay	analytical	protein	kinetics
Protein complementation	two-hybrid	perspective + analytical	protein	*in vivo*
	FRET	analytical	protein	*in vivo* location
Array technologies	protein array	perspective	protein	
	pep spot	perspective + analytical	residues	
	phage display	perspective	residues	

[a] *Perspective* applies to methods that have currently been used to explore uncharacterized protein interaction, whereas *analytical* applies to experimental procedures that investigate known interactions. Note that the resolution level is meant to be general. When mutants, etc., are used, the resolution of every method can be at the residue level.

[b] Also called coIP or protein complex purification. Note that copurifications are usually combined with subsequent mass spectrometric analysis of the copurified proteins.

[c] Far Western blots work like ordinary Western blots except that a blotted protein is detected by an interacting proteins as opposed to an antibody. The interacting protein can be labeled itself or detected by a labeled antibody.

[d] In cross-linking experiments proteins are cross-linked by bifunctional chemicals so that they cannot fall apart when purified. This allows more stringent purification protocols to be applied and thus cleaner preparations are achieved that can be analyzed more easily for their protein content.

found interactions as transient or stable, it is clear that the methods detect different kinds of interactions and are therefore highly complementary (Figure 2).

Table 2 High-throughput interaction detection experiments

Organism	No. purified complexes	No. binary interactions	Method	Reference
H. pylori	–	1465	two-hybrid pooling of fragment library	105
E. coli	530	5254[a]	tap tag coIP	21
P. falciparum	–	2846	two-hybrid pooling approach	80
S. cerevisiae	–	4549	two-hybrid pooling approach	62
S. cerevisiae	–	1511	two-hybrid array approach	132
S. cerevisiae	589	3757[a]	tap tag coIP	41
S. cerevisiae	741	2583[a]	flag tag coIP	58
C. elegans	–	4624	two-hybrid pooling approach	85
D. melanogaster	–	20676	two-hybrid pooling approach	45
D. melanogaster	–	$\sim$ 2300	two-hybrid pooling approach	37
H. sapiens	32	1814[a]	tap tag coIP	17
H. sapiens	–	2800	two-hybrid array approach	111
H. sapiens	–	3186	two-hybrid array approach	123

[a] Binary interactions are derived from co-purified complexes only as a set of bait–prey pairs, according to the "spoke model" (Figure 5).

Y2H screens and large-scale complex purifications have now generated large data sets (Table 2). Together with manually curated small-scale data, these interactions can now be downloaded from several protein interaction databases (Table 3).

3 Validation of Experimental Protein–Protein Interaction Data

Any experimental method suffers from a certain number of false positives and false negatives. However, high-throughput methods are more prone to such artifacts as they generate them as systematically as they generate valid data. Several computational methods have been proposed to evaluate the quality of interaction data. Critical for any method is the benchmark data set that is a "gold standard" of interactions that can be considered as reliable and compared to a new interaction data set. With such benchmarks most methods can suggest rough estimates of the rate of false negatives/positives. Some important approaches and benchmarks are summarized in the following sections.

3.1 Crystal Structures as Benchmarks

The crystal structure of a protein complex is the "gold standard" for a protein interaction as the structure provides the most detailed information. Unfortunately, there are not many structures available for protein complexes consisting of three or more proteins. One of the best studied cases is the crystal structure of yeast RNA polymerase II that consists of 10 subunits which are connected by 18 interactions [25]. While Y2H studies of a similar complex (RNA polymerase III and several associated proteins) found only 12 interactions among the 19 proteins [36], the crystal structure of RNA polymerase II shows a number of "interactions" where subunits barely touch each other. It is unlikely that such weak interactions will be detected by any method except by structure analysis. Of course, the polymerase complex interacts with many other associated proteins that are not detected when the purified complex is analyzed by MS because they are lost upon purification. Edwards and coworkers [30] estimated the false negative rate of various methods to be between 0% (for chemical cross-linking) and 67% (for Far Western experiments) based on the crystal structure of RNA polymerase II (which was treated as a "true gold standard"). Similarly, their estimates for false positives ranged from 41% (for chemical cross-linking) to 67% (Far Western).

3.2 Overlap with Protein Complex Data

In addition to structural information, data from purified complexes can be used for evaluation purposes as well. Edwards and coworkers [30] analyzed the overlap of various interaction data sets with the MIPS complex catalog [92] – a set of individually validated protein complexes. Based on the overlap with this data set, these authors estimated the rate of false negatives to be between 51 and 85% for various high-throughput two-hybrid data sets and to be 50% for high-throughput complex purification data. Note, however, that estimates of false positives and negatives heavily depend on the precise methods used, the biological object and the bioinformatics filtering applied to the raw data.

3.3 Correlation with Expression Data

Deane and coworkers [28] used expression data to measure the overall reliability of a given interaction data set. The idea is to take interacting proteins and see if their expression is coregulated, i.e. if interactions correlate with expression levels. Deane called their measure *expression profile reliability* (EPR) index. It compares the RNA expression profiles for the proteins whose interactions are found in a screen with expression profiles for known interacting and noninteracting pairs of proteins. Based on such correlation studies, Deane and coworkers estimated the error rates to be roughly 50% among the Y2H interactions in the Database of Interacting Proteins (DIP) [115]. Note that most such correlations can only evaluate *data sets* as opposed to *single interactions*. Similar studies involving correlations of expression data and protein interaction data were published by Ge and coworkers [42], Grigoriev [50], and Tornow and coworkers [128].

3.4 Functional Annotation

Interacting pairs can be also evaluated by means of their annotation in databases, e.g. Gene Ontology (GO) terms [44]. If two interacting proteins have the same annotation, e.g. "vesicular transport", this supports the validity of their interaction. This method was used by Sprinzak and coworkers [121] to validate interaction data. Of course, any other type of annotation can be used in similar ways. Other studies broadened this concept by not limiting it to annotations in databases, but by using "keyword retrieval" in general, e.g. from PubMed abstracts (see Section 4.5). However, even truly related proteins show only a partial keyword overlap and sometimes none at all.

3.5 Localization

Proteins can only interact if they occur in the same subcellular compartment, for example the nucleus. Sprinzak and coworkers [122] used functional annotation and localization data to estimate the false-positive rate among high-throughput two-hybrid data to be on the order of 50%.

3.6 Paralogous Proteins and Evolutionary Rate

The paralogous verification method (PVM) of Deane and coworkers [28] judges an interaction likely if the putatively interacting pair has paralogs that also interact. In contrast to the EPR index (see Section 3.3), which evaluates data sets of interactions, PVM scores individual interactions. On a test set, PVM identified correctly 40% of true interactions with an estimated false-positive rate of about 1%.

Fraser and coworkers [38, 39] suggested that the correlation between protein interaction and evolutionary rate may allow one to use sequence comparisons to statistically assess the quality of interaction data sets (see Section 8.2.1 for more details).

3.7 Other Approaches

There are a number of other criteria that can be used to evaluate interactions. In fact, any kind of information can be used that reflects similarities between putative interaction partners. In bacteria, genes that are *neighbors* in the genome are often organized in the same operon and thus often function together [26]. Similarly, protein pairs that are conserved throughout evolution may have been coselected because they are required for a common function and thus may interact. Pairs or groups of proteins with such shared evolutionary patterns are said to have common *phylogenetic profiles* [102] (see also Chapter 32). Finally, *genetic interactions*, i.e. mutations in two or more different genes that show a stronger phenotype any of the individual genes support a physical interaction between them or, in fact, the other way round [73, 142]. Note that genetic interactions provide rather indirect evidence that two proteins interact physically as opposed to biochemical or physical methods. Goldberg and Roth [49] exploited the *neighborhood cohesiveness* property of small-world networks (see Section 5.4), to assess confidence for individual protein–protein interactions. By ascertaining how well each protein–protein interaction fits the pattern of a small-world network, they were able to stratify even those interactions with identical experimental evidence. Another simple measure has been introduced by Saito and coworkers [112, 113]. Their *interaction generality* measure is basically the number of proteins involved in a given interaction. Saito and coworkers found that interactions with low generalities are more likely to be reproducible in other independent assays. However, this strategy appears to work only for raw interaction data that have not been filtered for "sticky" proteins (i.e. proteins that appear to have many unspecific interactions).

3.8 Combined Approaches

Ideally, several different sources of information should be combined to evaluate interaction data, particularly high-throughput data. This approach was chosen by von Mering and coworkers [137], who demonstrated that the number of false-positive interactions can be reduced by focusing on the intersection of interactions generated by different kinds of experimental technologies [e.g. the overlap between Y2H interactions and coimmunoprecipitation (coIP) data]. Furthermore they showed that such overlapping data mainly consist of interactions in which both partners have the same functional annotation and cellular localization. Similar studies have been carried out by Sprinzak and coworkers [122] (who combined localization and annotation data) and other authors [9]. Figures 3 and 4 show examples of data quality evaluations and how to improve them by integrating several sources.

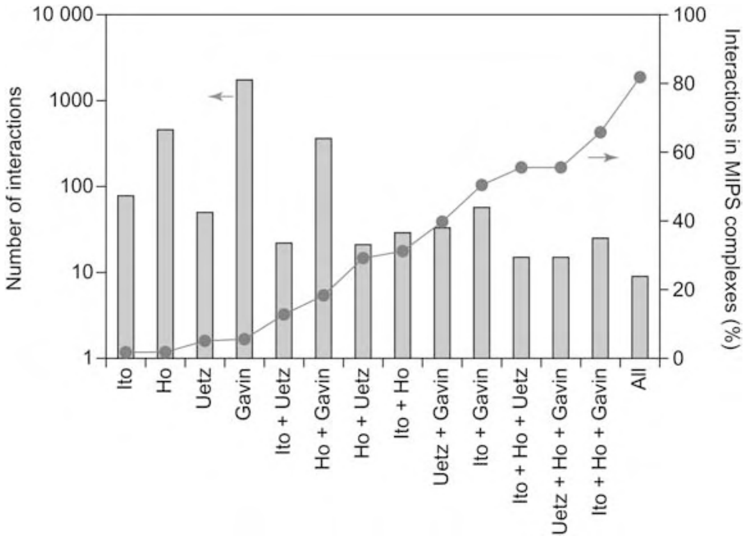


Figure 3 The size of the different genome-wide data sets and their possible intersections and their consistency with the MIPS complexes catalog [92]. The bars (relating to the left y-axis) indicate the number of interactions in each individual data set and each possible intersection of the data sets (e.g. 'Ito + Uetz' contains only interactions that are both within Ito and Uetz). The line (relating to the right y-axis) shows what fraction of these interactions overlaps with protein pairs within the same MIPS complex. The individual data sets are arranged on the left, pairwise intersections of data sets in middle and higher-order intersections (three or more data sets) on the right. As the degree of intersection among the data sets increases, the fraction of interactions within the same MIPS complex increases. The different data sets are complementary and cover more interactions than each data set individually. (From Ref. [30].)

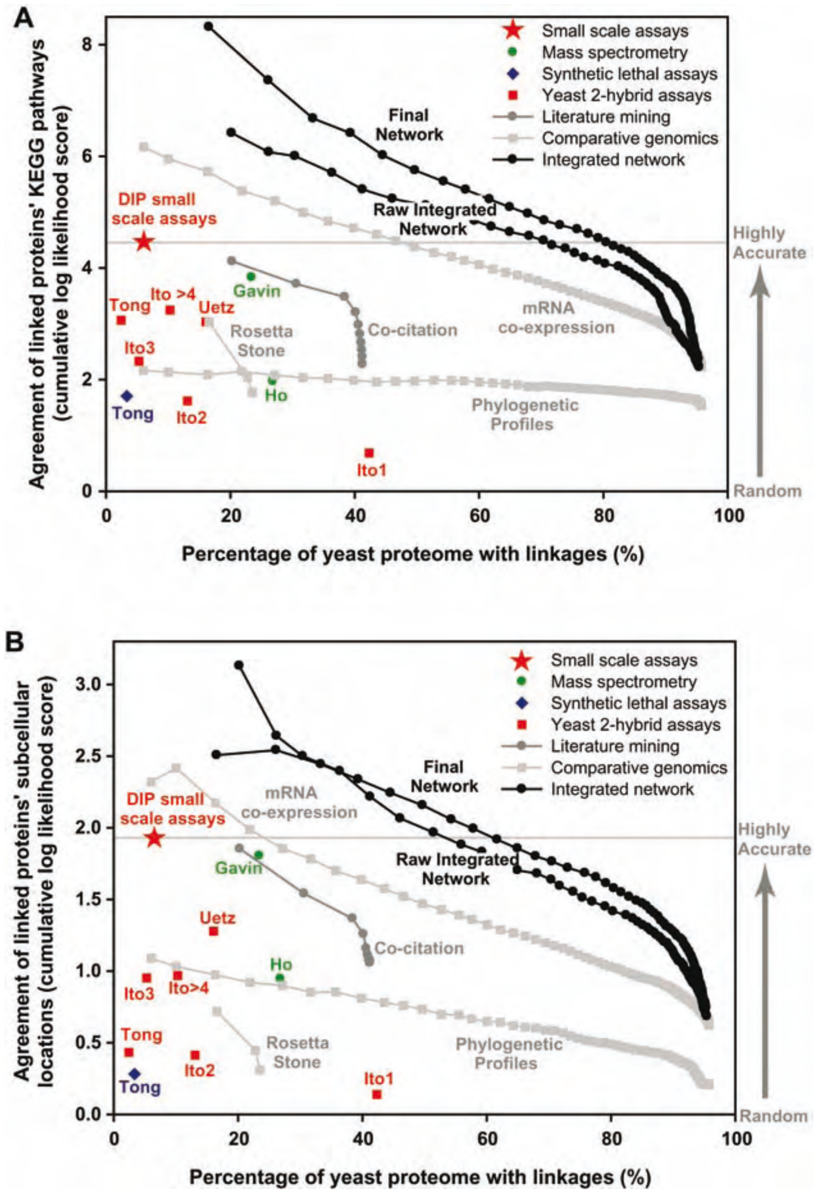
However, as long as automated validation methods are not significantly improved, validation by manual expert analysis remains key.

3.9 Comparison of Specific Data Sets

A lot of effort has been spent on comparing the quality of different data sets [30]. Such comparisons also serve as cross-validation as shown by the following example (data from Ref. [24]).

3.9.1 **Comparison of Tandem Affinity Purification (TAP) and High-throughput MS (HMS) complex purification data**

Protein complex purification has been used most extensively in yeast and two main methodological variations have been used: TAP [41] and HMS [58] protein complex identification (HMS-PCI). The two approaches differ in the way proteins were expressed (from their natural chromosomal location in TAP



and from overexpressing plasmids in HMS-PCI) and in the tags that were used for the purification of the protein complexes. Such technical details can cause tremendous differences in the resulting interaction data.

On average, the number of proteins common to the TAP and HMS-PCI data sets is less than 9% of the total number of proteins in both data sets. For example, employment of Yju2p as a bait identified 15 proteins using TAP

Figure 4 Benchmarked accuracy and extent of functional genomics data sets and the integrated networks. A critical point is the comparable performance of the networks on distinct benchmarks, which assess the tendencies for linked genes to share (A) KEGG pathway annotations [70] and (B) protein subcellular locations [59]. Each x-axis indicates the percentage of protein-encoding yeast genes provided with linkages by the plotted data; each y-axis indicates relative accuracy, measured as the agreement of the linked genes' annotations on that benchmark. The "gold standards" of accuracy (red star) for calibrating the benchmarks are small-scale protein–protein interaction data from DIP [115]. Colored markers indicate experimental linkages; gray markers, computational. The initial integrated network (lower black line), trained using only the KEGG benchmark, has measurably higher accuracy than any individual data set on the subcellular localization benchmark; adding context-inferred linkages in the final network (upper black line) further improves the size and accuracy of the network (see Ref. [81] for details and additional benchmarks).

and 15 using HMS-PCI. Only one protein (Prp19p) is common to both sets. This shows that experimental details have to be borne in mind when interaction data sets are compared. As another example, there can be considerable disparity between the size of complexes generated by TAP and HMS-PCI. Complexes generated using baits Pwp2p and Kap104p contain 54 and four proteins, respectively, using TAP, compared to seven and 36 using HMS-PCI.

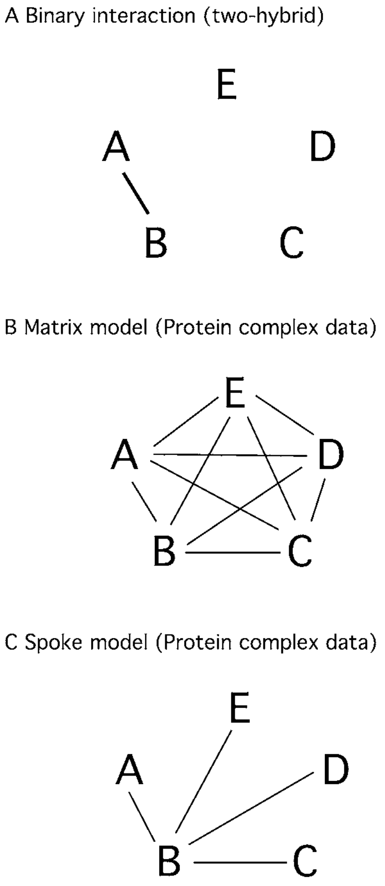
3.9.2 Comparison between Y2H and MS data sets

The largest overlap found by Cornell and coworkers was between the TAP and the Uetz Y2H data sets [41, 132] where 21% of the interactions found by Y2H are supported by affinity purification. In contrast, less than 7% of the Y2H interactions in the Ito data set are supported by TAP.

3.9.3 Comparison of Spoke versus Matrix Models

Cornell and coworkers [24] compared TAP and HMS-PCI complexes using the spoke and matrix models (Figure 5). The two data sets were validated by counting coexpressed proteins and by comparing functional annotations of all proteins in a complex. For example, a protein complex is considered to be of high quality if all proteins in it are expressed in the same cell at similar levels and if all proteins have similar functional annotations (Figure 6). According to Cornell et al. [24], the spoke model generated 3163 protein pairs from TAP complexes and 3503 pairs from HMS-PCI. The matrix model generates 17 281 protein pairs from TAP complexes and 30 672 pairs from HMS-PCI. Interestingly, Cornell and coworkers [24] found that the experimental approach (TAP versus HMS-PCI) affected their analysis more than the choice of data model (here, matrix versus spoke model). This shows that data sets may be more important than data models!

Thus, the *choice of data sets* is often critical. This is also true for subsets of proteins which can be selected by a wide range of fairly arbitrary criteria. For


A Binary interaction (two-hybrid)

B Matrix model (Protein complex data)

C Spoke model (Protein complex data)

Figure 5 Data models for protein interactions in complexes. (A) A protein complex may consist of five subunits, A–E, hold together by five interactions. A two-hybrid assay may detect only interaction A–B, while other methods may detect the other interactions (dotted lines). (B) The matrix model assumes that each subunit interacts with every other subunit. (C) The spoke model assumes that the bait protein interacts with all other proteins.

example, a protein pair that occurs in more than one complex can be defined as a frequently observed pair (FOP), while those that occur only once may be defined as singly observed pairs (SOPs). Furthermore, those SOPs in which each protein occurs only once in that data set have been defined as unique SOPs (U-SOPs). Analysis of expression profile correlation shows that FOPs tend to have much greater correlation coefficients than SOPs. The correlation coefficients for SOPs are similar to those of random protein pairs [24].

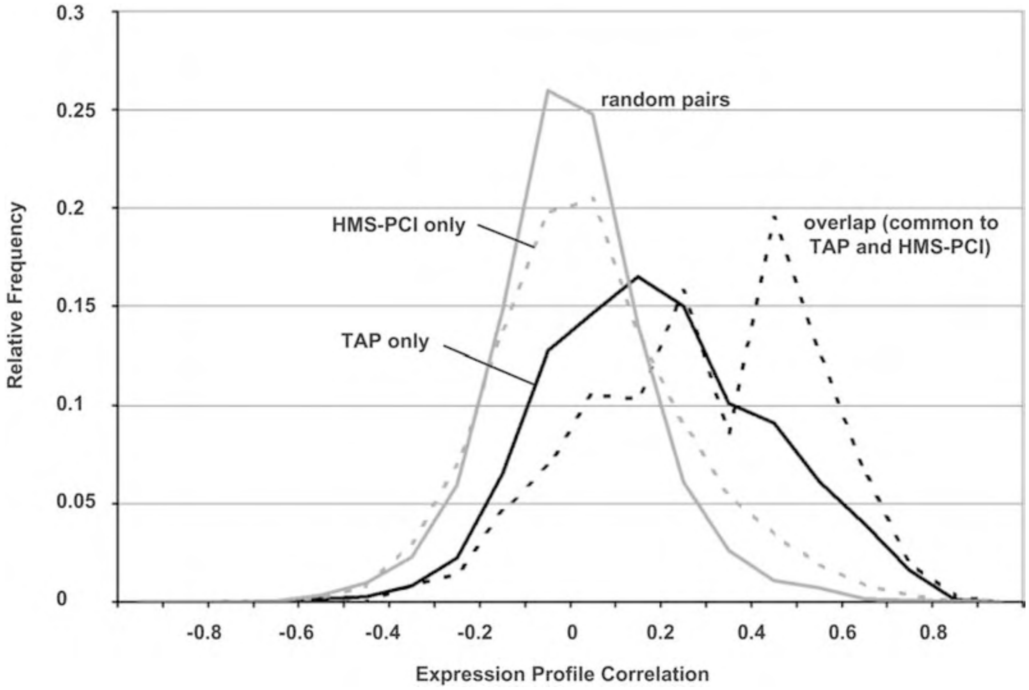


Figure 6 Frequency distribution of expression profile correlations for pairs of proteins in affinity-purified protein complexes, purified using the same bait protein by TAP and HMS-PCI. Key: black dashed line, overlap – pairs of proteins common to both TAP and HMS-PCI complexes; grey dashed line, HMS-PCI only – pairs of proteins purified only by HMS-PCI; black solid line, TAP only – pairs of proteins purified by TAP only; grey solid line, random pairs – a set of 4010 pairs of randomly chosen proteins [24].

4 Predicting Protein–Protein Interactions

Predicting interactions is conceptionally similar to validation. While validation involves the comparison of interaction data to certain benchmarks such as colocalization data, such criteria can also be used to predict protein–protein interactions. In other words, *potentially interacting* proteins can be selected from the pool of *all possible* protein pairs of a genome by applying such filtering criteria.

The maximum number of possible protein–protein interactions in a proteome is approximately:

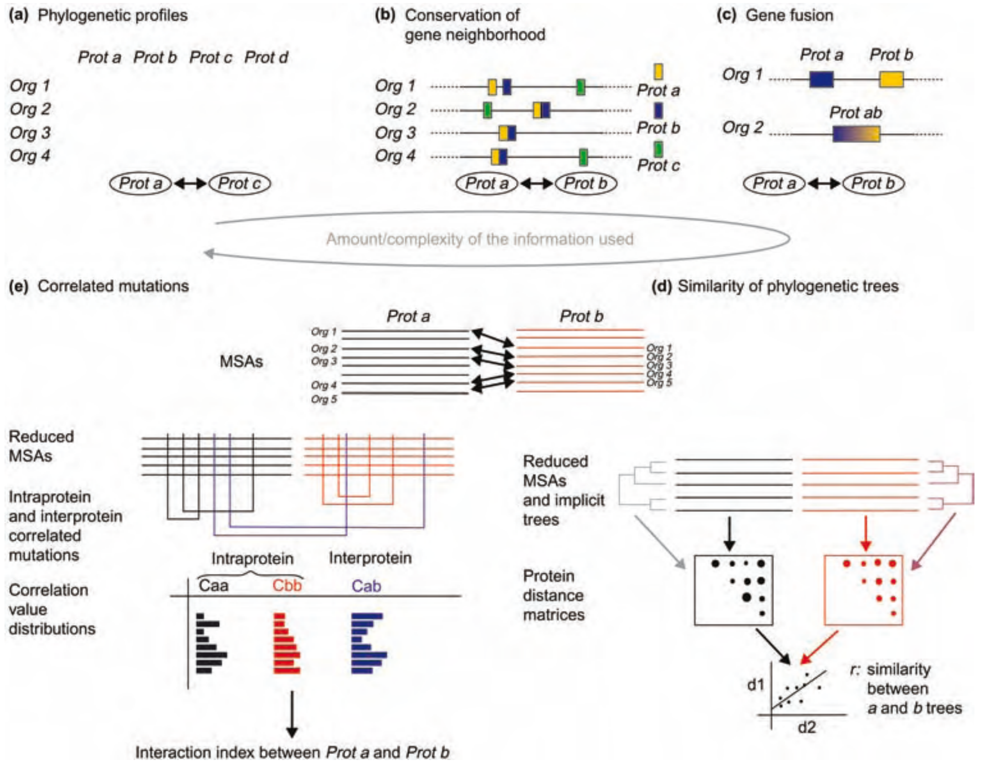
$$\binom{n+k-1}{k} = \frac{(n+k-1)!}{k!(n-1)!}$$

with n being the size of the proteome (i.e. the number of proteins) and $k = 2$ for binary interactions including homodimers, and counting an interaction

Figure 7 Methods for predicting protein interaction partners from genomic and sequence information. The methods are presented according to the amount of information they include, ranging from simple patterns of gene presence in genomes to detailed sequence information (amino acids in each position) in protein families. (a) Phylogenetic profiles [102]. A profile is constructed for each protein (Prot a–Prot d), recording its presence (1) or absence (0) in a set of organisms (Org 1–Org 4). Pairs of proteins with identical (or similar) phylogenetic profiles are predicted to interact (Prot a and Prot c in this case). (b) Conservation of gene neighborhood [26]. Proteins whose genes are physically close in the genomes of various organisms are predicted to interact (Prot a and Prot b). (c) Gene fusion [31, 89]. Two proteins of a given organism (Prot a and Prot b of Org 1) are predict to interact if they form part of a single protein in other organisms (Org 2). (d) Similarity of phylogenetic trees (mirrortree) [46, 101]. To obtain a quantitative indicator of the interaction between two proteins (Prot a and Prot b), the multiple sequence alignments (MSAs) of both proteins are reduced to the set of organisms common to the two proteins (Org 1–Org 5). Each of the reduced alignments is used to construct the corresponding intersequence distance matrix. These matrices are commonly used to construct the corresponding phylogenetic trees. Finally, the linear correlation between these distance matrices is calculated. High correlation values are interpreted as indicative of the similarity between phylogenetic trees and hence are taken as predicted interactions. (e) Correlated mutations (i2h) [100]. The first step (reduction of the MSAs to a set of common organisms) is the same as that described for the mirrortree method (d). A correlation coefficient is calculated for every pair of residues. The pairs are divided into three sets: two for the intraprotein pairs (Caa and Cbb; pairs of positions within Prot a and within Prot b) and one for the interprotein pairs (Cab; one position from Prot a and one from Prot b). The distributions of correlation values are recorded for these three sets. The “interaction index” is calculated by comparing the distribution of interprotein correlations with the two distributions of intraprotein correlations [100].

pair A–B and B–A as a single interaction. For example, a proteome of a small prokaryote ($n = 1000$) corresponds to a maximum of 500 500 potential binary interactions.

Note that interacting protein pairs may behave differently depending on their “orientation” in an experiment. For example, when an antibody against protein A precipitates protein B, an antibody against protein B may not precipitate protein A. For that reason, many experiments have to be carried out in both “orientations” for full coverage. This leads to exponentially growing numbers in experiments required for genome-wide studies. For example, while a small bacterium with 1000 genes has 10^6 interaction pairs to be tested, a medium-sized eukaryotic genome requires already 100×10^6 pairwise combinations to be tested. Given that the number of binary protein combinations grows exponentially it is unlikely that the interactomes of higher eukaryotes will be fully covered by experimental means in the near future. Note that eukaryotes also exhibit variations such as alternative splicing or post-translational modifications which often affect protein interactions and thus increase complexity by at least another order of magnitude. In addition, all experimental methods show a certain degree of false negatives so that


(a) Phylogenetic profiles
Prot a Prot b Prot c Prot d
Org 1
Org 2
Org 3
Org 4
Prot a ↔ Prot c
(b) Conservation of gene neighborhood
Org 1
Org 2
Org 3
Org 4
Prot a
Prot b
Prot c
Prot a ↔ Prot b
(c) Gene fusion
Prot a
Prot b
Org 1
Prot ab
Org 2
Prot a ↔ Prot b
Amount/complexity of the information used
(e) Correlated mutations
(d) Similarity of phylogenetic trees
Prot a
Prot b
MSAs
Org 1
Org 2
Org 3
Org 4
Org 5
Reduced MSAs
Intraprotein and interprotein correlated mutations
Intraprotein
Interprotein
Correlation value distributions
Caa
Cbb
Cab
Interaction index between Prot a and Prot b
Reduced MSAs and implicit trees
Protein distance matrices
d1
d2
r: similarity between a and b trees

even experiments rarely identify all interactions. The experimental bottleneck motivated the design of algorithms for predicting protein interactions (Table 3 lists databases of predicted functional associations). It is usually much more efficient to verify predicted interactions experimentally than just doing experiments randomly without such preselection. One has to bear in mind, however, that predictions are often based on experimental limitations. For example, interactions among membrane proteins are usually underpredicted because they are also underrepresented in the data sets that are used to make the predictions.

One can distinguish three main types of algorithms:

(i) Predictions based on functional relationships such as colocalization, genomic context, etc. Text mining is a special case of such relationships where protein interactions may be already encoded in the literature.

(ii) Docking of known protein structures.

(iii) Homologous interactions in organism A can be predicted based on existing interaction data in organism B and *vice versa*.

Table 3 Databases and information resources for protein interaction data (this list is available online at http://uetz.fzk.de)

Name	Full name and/or description	URL
General interaction repositories		
BIND	biomolecular interaction network database	http://www.bind.ca
DIP	database of interacting proteins: experimentally determined protein–protein interactions	http://dip.doe-mbi.ucla.edu
IntAct	open source database system and analysis tool for protein interaction data	http://www.ebi.ac.uk/intact
Species specific interaction databases		
BioGRID	database of protein and genetic interactions of most eukaryotic model organisms	http://www.thebiogrid.org
MINT	database of biomolecular interactions in mammalian proteomes	http://mint.bio.uniroma2.it/mint
hp-DPI	database of protein interactions in *H. pylori*	http://dpi.nhri.org.tw/hp
MPPI	MIPS mammalian protein–protein interaction database	http://mips.gsf.de/proj/ppi
MPact	MIPS protein interaction resource on yeast	http://mips.gsf.de/genre/proj/mpact
CYGD	protein–protein interactions section of the Comprehensive Yeast Genome Database	http://mips.gsf.de/proj/yeast/CYGD/interaction
HIV Interactions	interactions between HIV and host proteins	http://www.ncbi.nlm.nih.gov/RefSeq/HIVInteractions
HPRD	human protein reference database: domains, modifications, diseases	http://www.hprd.org
HPID	human protein interaction database	http://www.hpid.org
PPI Viewer	protein interactions in mouse	http://fantom21.gsc.riken.go.jp/PPI
Databases of predicted functional associations		
STRING	predicted functional associations between proteins	http://string.embl.de
FusionDB	database of bacterial and archaeal gene fusion events	http://igs-server.cnrs-mrs.fr/FusionDB
ALLFuse	database of functional associations of proteins in complete genomes	http://cgg.ebi.ac.uk/services/allfuse
Prolinks	database of protein functional linkages derived from coevolution	http://dip.doe-mbi.ucla.edu/pronav
Predictome	predicted functional associations and interactions	http://predictome.bu.edu

Table 3 (continued)

Name	Full name and/or description	URL
Bioverse	structural, functional, and contextual annotations of proteins and proteomes	http://bioverse.compbio.washington.edu
InterWeaver.	protein interaction predictions based on various evidence	http://interweaver.i2r.a-star.edu.sg
GeneNet	database on gene network components	http://wwwmgs.bionet.nsc.ru/mgs/gnw/ genenet
OPHID	online predicted human interaction database	http://ophid.utoronto.ca/ophid
POINT2005	prediction of human protein–protein interactome	http://point.bioinformatics.tw/intro/intro.jsp
VisANT	bio-network visualization and analysis tool	http://visant.bu.edu
PLEX	Protein Link Explorer; constructs phylogenetic profiles	http://bioinformatics.icmb.utexas.edu/plex
Structural databases		
3DID	3-D structures of interacting domains	http://3did.embl.de
DDIB	database of domain interactions and binding	http://www.ddib.org
Inter-Chain	β—sheets protein– protein interactions mediated by interchain β-sheet formation	http://www.igb.uci.edu/servers/icbs
InterDom	putative protein domain interactions	http://interdom.lit.org.sg
PDZBase	protein–protein interactions involving PDZ domains	http://icb.med.cornell.edu/services/pdz/start
PSIbase	interaction of proteins with known 3-D structures	http://psibase.kaist.ac.kr
InterPreTS	tool to predict protein interactions using 3-D information	http://speedy.embl-heidelberg.de/people/patrick/interprets/index.html
PPI Server	analysis of protein–protein interfaces of protein complexes from PDB	http://www.biochem.ucl.ac.uk/bsm/PP/server
3D-Dock Suite	3-D DOCK predicts the binding geometry of two biomolecules	http://www.bmm.icnet.uk/docking
PPI interfaces	interacting residues in protein–protein interfaces in PDB	http://home.ku.edu.tr/∼okeskin/INTERFACE/INTERFACES.html
Het-PDB Navi	hetero-atoms in protein structures (a database for protein–small molecule interactions)	http://daisy.nagahama-i-bio.ac.jp/golab/hetpdbnavi.htm
Location databases		
PINdb	proteins interacting in nucleus (human and yeast)	http://pin.mskcc.org
PSORTdb	a database of protein subcellular localizations for bacteria	http://db.psort.org/docs/documentation.html

Table 3 (continued)

Name	Full name and/or description	URL
Signaling Pathway Databases		
aMAZE	annotation, management and analysis of biochemical and signaling pathways	http://www.amaze.ulb.ac.be
BioCarta	online maps of metabolic and signaling pathways	http://www.biocarta.com/genes/ allPathways.asp
BRITE	biomolecular relations in information transmission and expression, part of KEGG	http://www.genome.ad.jp/brite
Reactome	a knowledgebase of biological pathways	http://www.reactome.org
DRC	database of ribosomal crosslinks	http://www.mpimg-berlin-dahlem.mpg.de/∼ag_ribo/ag_brimacombe/drc
ROSPath	reactive oxygen species signaling pathway	http://rospath.ewha.ac.kr
STCDB	signal transductions classification database	http://bibiserv.techfak.uni-bielefeld.de/stcdb
pSTIING	regulatory networks relevant to chronic inflammation, cell migration and cancer	http://pstiing.licr.org
Kinetics		
KDBI	kinetic data of protein–protein, protein–nucleic acid and ligand–nucleic acid binding.	http://xin.cz3.nus.edu.sg/group/kdbi/kdbi.asp
PINT	Protein–protein Interactions Thermodynamic Database	http://www.bioinfodatabase.com/pint/index.html
Other links		
Jena Link list	protein interaction link list at the Jena Centre for Bioinformatics, Germany	http://www.imb-jena.de/jcb/ppi
Pawson Lab	information on protein interaction domains	http://www.mshri.on.ca/pawson/domains.html

4.1 Predictions Based on Genomic Context

Several algorithms predict protein associations on the basis of sequence data from completely sequenced genomes and are inspired by comparative genomics techniques (for details see Chapter 32). The main methods are as follows (Figure 7).

4.1.1 The Rosetta Stone Method

This method is based on the fact that multi-domain proteins found in one organism may be split in another. It is thus likely that the fused domains interact within the multi-domain protein and thus the separate domains may interact as well [89]. Illuminating in this respect are some observations by Aloy and coworkers [4] who found that SH2 and SH3 domains within a single multi-domain protein can "interact" in at least three different ways in a protein crystal. This indicates that the two domains do not always interact in the same way, i.e. their interaction surfaces are not strictly conserved. It is equally possible that they have been oriented in different ways in the crystal because there are equally different ways to crystallize.

Interestingly, Aloy and coworkers [4] also found several different classes of successful Rosetta predictions – fused domains can interact similarly or differently when compared to their separated counterparts. For example, the enzyme imidazole glycerophosphate synthase is composed of two different structural domains – a histidine biosynthesis domain and a class I glutamine amido-transferase domain. These domains are encoded by separate genes in archaea but fused in eukarya. Despite the huge evolutionary distance between these organisms, the interaction is conserved. However, for other fusions the interaction is different. For example, despite significant sequence similarity (PID = 34%, where PID is the percentage sequence identity, i.e. the number of identical residues divided by the number of structurally equivalent residues), the FAD/NAD(P) binding and thioredoxin-like domains interact differently when separated.

4.1.2 **Gene Neighborhood**

The gene neighborhood approach rests on the fact that many functionally related genes in bacteria are organized in operons, i.e. they are gene neighbors. Furthermore, often proteins encoded in one operon interact in a protein complex, e.g. a multi-protein enzyme complex. Neighboring genes in bacteria are therefore more likely to interact than proteins encoded in other regions of the chromosome. Unfortunately this is not true for eukaryotes which usually do not have operons.

4.1.3 **Phylogenetic Profiles**

The phylogenetic profile method [102] deduces functional links between genes that have similar occurrence patterns of orthologs in a set of reference genomes. In other words, if pairs or groups of proteins are maintained in many different genomes this may be because they have been selected as groups and thus may functionally or physically interact. Phylogenetic links are collected by the STRING database ([136] and see also Chapter 32) and other websites listed in Table 3.

The Protein Link EXplorer (PLEX) is a web-based environment that allows the construction of a phylogenetic profile for any given amino acid sequence, and its comparison with profiles of approximately 350 000 predicted genes from 89 genomes [27].

4.1.4 Similarity of Phylogenetic Trees (SPT)

As we have seen, interacting protein pairs coevolve, e.g. insulin and its receptors. In such cases, the corresponding phylogenetic trees of the interacting proteins show a greater degree of similarity (symmetry) than those of non-interacting proteins. Thus, similar phylogenetic trees of potential interactors support their interaction. In fact, the phylogenetic profile method is related to this procedure, but goes even one step further because gain or loss of a gene is the ultimate step in protein evolution. In other words, while phylogenetic profiles look for the absence or presence of homologous proteins, the SPT method quantifies homology. The SPT method is also called the *mirrortree* method [101, 134].

4.1.5 *In Silico* Two-hybrid (I2H)

As a refinement of the previous method, the coevolution of interacting proteins can be followed more closely by quantifying the degree of covariation between pairs of residues from these proteins (correlated mutations). These positions may correspond to compensatory mutations that stabilize the mutations in one protein with changes in the other. The relationship between correlated residues and interacting surfaces has been used to predict interacting protein pairs based on the differential accumulation of correlated mutations between the interacting partners and within the individual proteins [100].

As in the case of the mirrortree method, the main limitation of the I2H approach is the need for complete alignments with a good coverage of species common to the two proteins under study.

4.2 Predictions Based on Known 3-D Structures

If the structures of all proteins were known, their interactions should be predictable by docking methods. Although docking of rigid structures is theoretically possible, the problem turned out to be much more difficult to solve because proteins are not rigid. In fact, many interactions do require some amount of induced fit for optimal binding [33, 48]. The problem and methods for solving it are discussed in more detail in Chapter 17.

4.3 Predicting Interaction Domains

Protein–protein interactions are usually mediated by specific protein domains. Some of the best-characterized domains are indeed interaction domains such as the SH2 or SH3 domains [84]. Several authors made use of this fact in order to identify interacting domains computationally [29, 95, 121]. Briefly, when certain proteins with a shared domain A interact with other proteins that share a domain B, this suggests that domains A and B mediate this interaction (Figure 8).

4.4 Predicting Homologous Interactions: Interologs

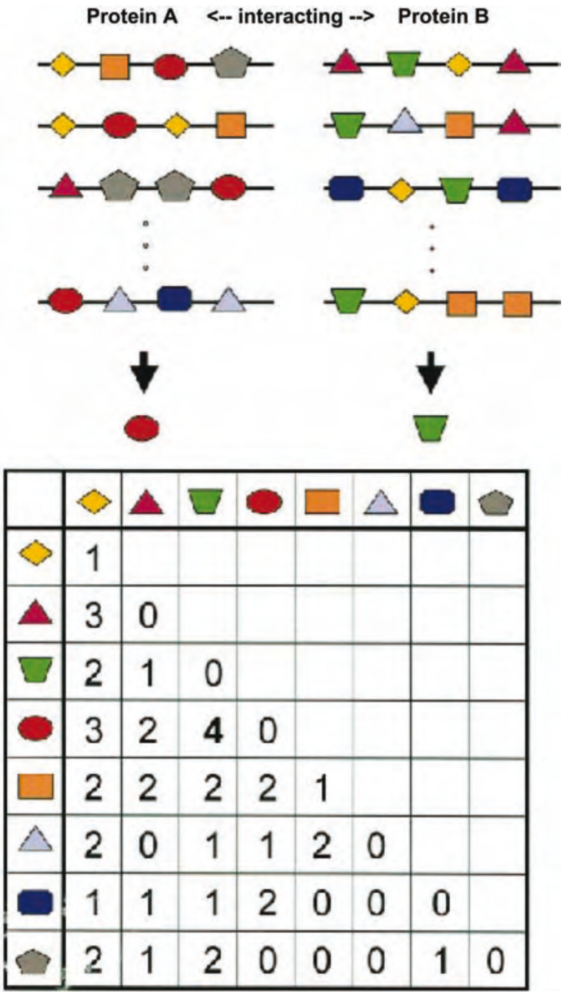
Proteins and their functions are usually well conserved throughout evolution. It has also been known for a long time that protein–protein interactions are conserved, e.g. in hemoglobins whose heterotetrameric structure is found throughout all vertebrates. Yu and coworkers [146] found that protein–protein interactions can be predicted when a pair of homologous proteins has a joint sequence identity above 80% or a joint E-value lower than 10^{-70}. These "joint "quantities are the geometric means of the identities or E-values for the two pairs of interacting proteins. This corresponds to an E-value lower than 10^{-10} for each orthologous protein and both having a sequence coverage of at least 80% of their residues. Similarly, Matthews and coworkers [91] used interaction data from yeast to predict interaction partners in *C. elegans*, many of which were indeed confirmed experimentally.

Thus, given that proteins A and B interact in yeast, we would predict that homologous proteins A′ and B′ in *Caenorhabditis elegans* interact as well (Figure 9). Such homologous, interacting pairs have been called *interologs* by Matthews and coworkers [91].

Interestingly, Matthews and coworkers [91] did not find a clear correlation between sequence similarity and the likelihood of an interaction being conserved between yeast and worm. However, this may be explained by different experimental systems and a small sample size. More importantly, given that interactions are usually mediated by protein domains or even short peptides, the *overall* similarity between interologs does not need to correlate with the propensity to interact.

Several studies used an interolog approach to predict human protein–protein interactions from model organism interaction data [19, 82, 110]. First, all reciprocal best BLAST hits were selected in the human and model organism proteomes (see Chapter 3 for details on BLAST). Then interaction data from yeast, worm and fly were mapped onto the human proteome. Using this approach, the three studies predicted between 23 889 and about 70 000 human interactions, respectively. The differences stem mostly from different starting interaction data sets and different cutoff criteria for protein similarity although Rhodes and coworkers [110] also used other criteria such as expression data and functional annotations.

There are not many attempts to predict interactions in bacteria as there are only a few systematic studies [21, 87, 105]. Wojcik and coworkers [140] predicted 1280 interactions in *Escherichia coli* based on two-hybrid data from



	◆	▲	⏢	⬬	■	△	▬	⬟
◆	1							
▲	3	0						
⏢	2	1	0					
⬬	3	2	4	0				
■	2	2	2	2	1			
△	2	0	1	1	2	0		
▬	1	1	1	2	0	0	0	
⬟	2	1	2	0	0	0	1	0

Figure 8 Computational identification of interaction domains. In the upper panel, each row contains the sequences of a pair of proteins (A,B) whose interaction was determined experimentally. Each sequence is characterized by its signatures, denoted here schematically by colored shapes. In the lower panel, a contingency table of the signature combinations is described, where each entry (i,j) in the table shows the number of protein pairs that contain signatures i and j in concert, i.e. where one protein contains signature i and its pair mate contains signature j. For example, the sequence-signature pair represented by an orange rectangle and a pink triangle appears in two pairs of interacting proteins. The most abundant pair of sequence-signatures is that of a red ellipse and a green trapezium which appears in four different pairs of interacting proteins. In the next step of the analysis the likelihood of the identified sequence-signature pairs is evaluated. (From Ref. [121].)

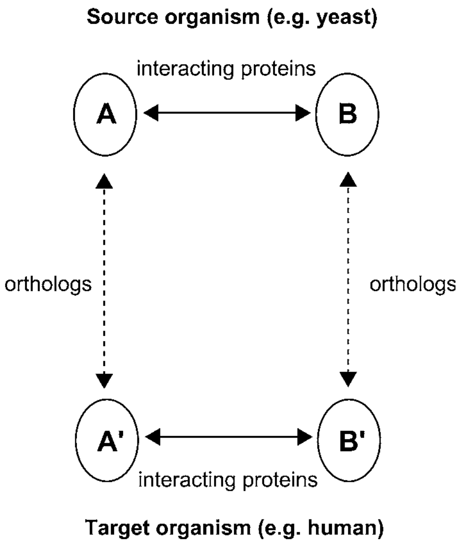


Figure 9 Protein–protein interologs. A–A′ and B–B′ are orthologs between the two organisms. Interolog mapping can be generalized when paralogs or whole protein *families* are considered (as opposed to single orthologs) (Modified after Ref. [146]).

Helicobacter pylori. However, they considered only 154 interactions to be reliable, based on their interacting domain profile pair (IDPP) method (Figure 10).

4.5 Predictions based on Literature Mining

Many protein–protein interactions are buried in the primary literature. In order to retrieve this information, Natural Language Processing (NLP) algorithms are used. These algorithms use abstracts and/or full texts of articles to extract links between genes and proteins. Knowledge extraction is not a trivial task as literature information is extremely noisy, due to insufficient synonym definitions, synonym variations and gene families with fuzzy naming conventions. Nevertheless, with the ongoing establishment of synonym databases and systematic names (HUGO [32], Entrez Gene [86]) for gene or gene product names, the predictions are getting more and more sophisticated. Jenssen et al. [66] analyzed over 10 million Medline titles and abstracts, and identified a cocitation network of almost 14 000 human proteins. Obviously, such relationships often do not represent physical interactions, e.g. when several oncogenes need to be mutated to cause cancer. Other attempts to mine Medline abstracts for protein interactions have been published by Marcotte and coworkers [90], Bunescu and coworkers [20], LaBaer [79], Oyama and

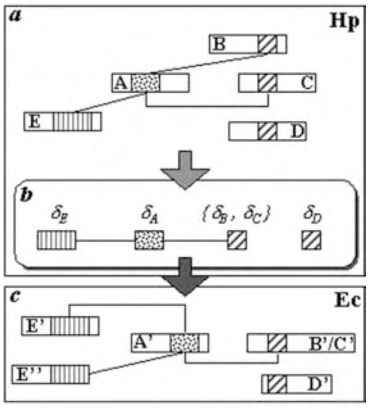


Figure 10 The IDPP method (p. 1143). An abstract domain cluster interaction map (b) was derived from the initial protein interaction map of *H. pylori* (Hp) (a). Domains were clustered together if (i) they shared significant sequence similarity and (ii) they shared a common interaction property with a third partner (e.g. interacting domains of proteins B and C both interact with A). Each domain or profile was then used as a probe for screening a library of *E. coli* (Ec) protein sequences and domain cluster interactions were transferred (c) (see Ref. [141] for further details).

coworkers [98], and Hao and coworkers [55]. More details on literature mining can be found in Chapter 34.

4.6 Validation of Predicted Protein–Protein Interactions

Even more than experimental techniques, predictive methods call for validation of their results and for assessment of their sensitivity and selectivity. In fact, validation of predicted protein interactions can be performed the same way as for experimental interactions (see Section 3).

Again, the first and most important way of validation is the comparison of predictions with some "gold standard" data set, i.e. manually or automatically collected data of high reliability, such as subsets of MIPS [92], DIP [115] or IntAct [56], or specifically collected data [137, 146]. The significance of the predictions is evaluated by calculating the fold improvement over a virtual random experiment and/or the correlation between the predicted and the standard data set.

Huynen and coworkers [61] compared the three methods based on genomic information. In their analysis, the method based on gene order could be applied to 37% of the *Mycoplasma genitalium* genes, whereas the phylogenetic profile method and the method based on gene fusion could only be applied to 11 and 6%, respectively. The combination of the three methods yielded predictions for 50% of *M. genitalium* genes, with just a small degree of overlap in the techniques. With respect to the accuracy of this test set, the percentage of pairs predicted by the three methods that either present a physical interaction, belong to the same macromolecular complex, form part of the same pathway or are implicated in the same process are: 78% for "gene fusion" (with no false positives), 80% for "conservation of gene order" and 63% for "phylogenetic profiles". The percentages for physical interactions only are 56, 30 and 23%, respectively.

5 Representing Protein–Protein Interactions as Graphs

Protein interactions within a cell form large networks and thus can be represented by graphs (Figure 11 and Ref. [12]) and be mathematically analyzed using graph theory [139]. Surprisingly, protein networks and many other natural networks are governed by a few simple organizing laws.

5.1 Graph Terminology

A graph G is a set of *vertices* (or *nodes* or *points*, here *proteins*) and *edges* (or *lines*, here *interactions)* denoted by $G = (V,E)$, where the elements of V are vertices and the elements of E are edges, two-element subsets of V, with $E \subseteq V \times V$. Often n is used to represent the number of vertices $|V|$ and m to represent the number of edges $|E|$. The usual way to picture a graph is by drawing a dot for each vertex and joining two of these dots by a line if the corresponding two vertices form an edge (Figure 12). How these points and lines are drawn is potentially relevant as the vertices represent different proteins and the interactions may differ in their nature.

It is the job of graph drawing algorithms to layout and display this information optimally. A graph is *undirected* if its edges are undirected, otherwise it is called *directed* (or a *digraph)*. Edges in a digraph are usually represented by arrowed lines. Vertices joined by an edge are said to be *adjacent*. A *neighbor* of a vertex v is a node adjacent to v. The *neighborhood* is the set of neighbors of vertex denoted by $N(v)$.

The *closed neighborhood* is the set of neighbors including v. A graph G is *complete* (or called a *clique)* if all its vertices are pairwise adjacent.

The *degree (or valency)* of a vertex v is the number of edges incident with v; this is equal to the number of neighbors of v. A vertex of degree 0 is *isolated*. In directed graphs one has to distinguish between the incoming degree, the number of edges ending at the node and the outgoing degree, the number of edges originating at the node.

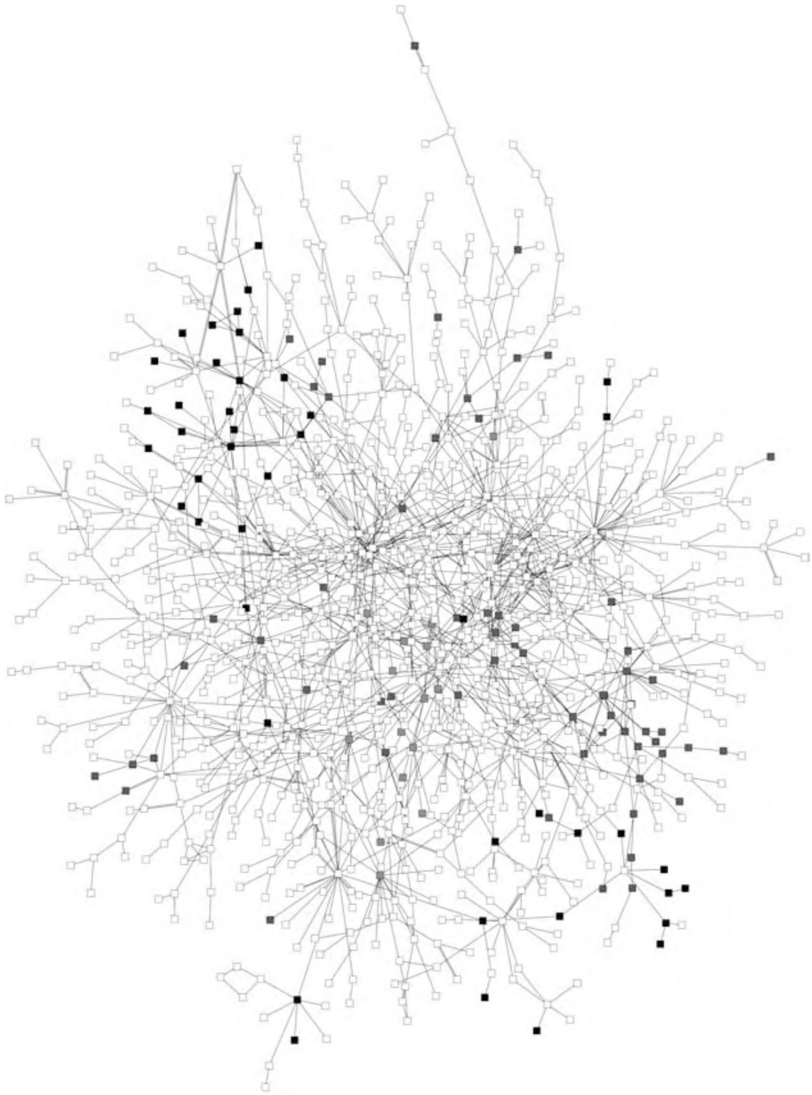


Figure 11 Yeast protein interaction network of around 1200 interacting proteins based on published interactions. Highlighted as dark nodes are cell structure proteins (a single functional class). Proteins in this category can be observed to cluster primarily in one region. Although interacting proteins are not depicted in a way that is consistent with their known cellular location (i.e. those proteins known to be present in the nucleus in the center of the interaction map and those present in plasma membranes in the periphery), signal-transduction pathways (or at least protein contact paths) can be inferred from this diagram. The graph was generated with the AGD software library (http://www.ads.tuwien.ac.at/AGD [130]).

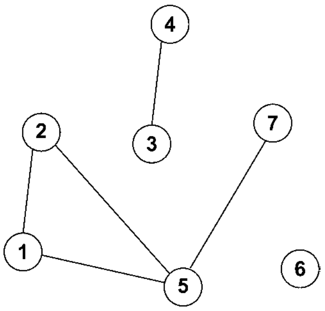


Figure 12 The graph on V = {1,2,3,4,5,6,7} with edge set E = {{1,2}, {1,5}, {2,5}, {3,4}, {5,7}}.

A *path* in a protein graph is a unique sequence of proteins and interactions starting and ending with a protein. The path *length* is the number of vertices in that sequence. A path which contains k vertices is commonly denoted by P_k. A graph G is called *connected* if any two vertices are linked by a path in G otherwise it is called *disconnected*. All known protein networks are disconnected because not all proteins belonging to a proteome are connected (maybe they are but we do not know all interactions).

The *distance* $d(u,v)$ in G between two vertices u, v is the length of a shortest u–v path in G. Here length is measured either by the number of edges on the path or, if each edge is weighted with a length, as the sum of the lengths of all edges on the path. The *diameter* of G is the greatest distance between any two vertices in G (although it is sometimes defined as the distance between two vertices averaged over all vertices). If a graph is disconnected, the diameter is equal to the maximum of the diameters of its connected components.

For an undirected graph the *clustering coefficient* of a vertex v is defined as $C_v = \frac{2 \cdot E_v}{k_v(k_v - 1)}$. It describes the ratio between E_v, the number of observed edges between the neighbors of v and the number of edges that could possibly exist between them. It is 1 if every neighbor connected to v is also connected to every other vertex within the neighborhood, and 0 if no vertex connected to v connects to any other vertex that is connected to v.

A directed graph with weighted edges is called a *network* (Figure 13). However, it should be noted that within network analysis, the definition of the term network may differ and may often refer to an undirected, unweighted graph. Actually, this is the case for most protein–protein interaction networks which often represent unweighted and undirected graphs, and which therefore strictly should be called protein–protein interaction graphs. Nevertheless, in this chapter we will also use the term network.

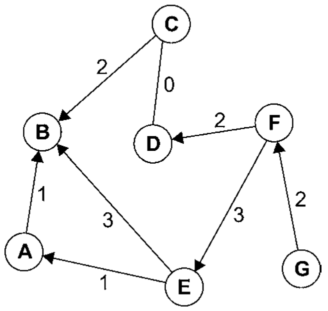


Figure 13 A simple network – a directed graph with weighted edges (*cf.* Figure 12). In protein–protein interaction networks directed edges are used to reflect protocols where one protein is used as a bait (arrow source) and associated proteins are identified as preys (arrow targets; especially in Y2H and protein complex purifications). Weights are used to numerically assess an interaction, for example the number of times it has been reproduced or its coexpression correlation, etc.

The most elementary global features of networks are:

1. The *connectivity distribution* (or *node degree distribution*) $P(k)$.
2. The *average path length* (or *average distance*) $\langle l \rangle = \frac{2}{n(n-1)} \sum_{u<v} d(u,v)$ defined by the average of the distances between any two vertices u, v within the network.
3. The *average clustering coefficient* $\langle C \rangle = \frac{1}{n} \sum_{i=1}^{n} C_i$ defined by the average of the clustering coefficients of all vertices within the network.

A detailed treatise on graph theory has been published by West [139]. A comprehensive online textbook about graph theory can be found at http://www.math.uni-hamburg.de/home/diestel/books/graph.theory. The MathWorld online encyclopedia provides more information on graph theory at http://mathworld.wolfram.com/topics/GraphTheory.html.

5.2 Network Models

Since the 1950s various network models have been proposed. Initially, random networks with no apparent design principles were constructed and described by Erdos and Renyi [107–109]. These authors simply assigned random edges to a set of randomly selected nodes. Later, other authors suggested rules for assigning edges and nodes resulting in nonrandom networks. Although metabolic networks have been known for decades, they have been treated as graphs only since the 1990s. Finally, Wagner [138] and Barabási and Albert [10, 11] introduced the concept of scale-free networks to protein networks. These models will be described in more detail below.

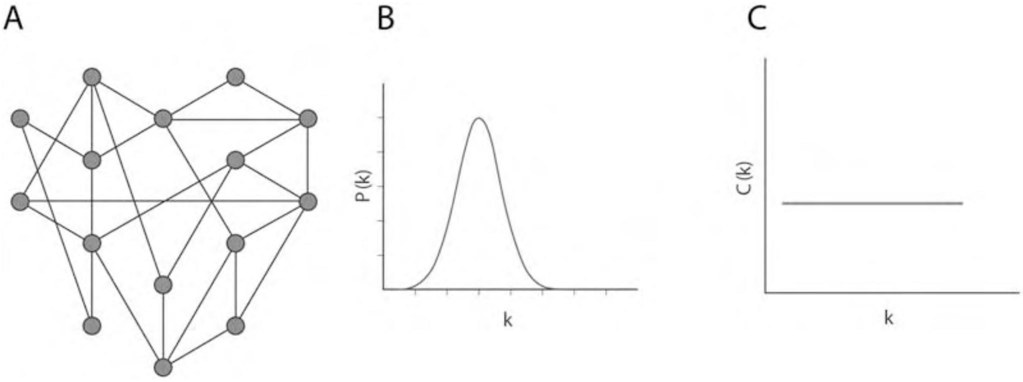


Figure 14 Random network. (A) small random network. The node degrees follow a Poisson distribution (B), which indicates that most nodes have approximately the same number of links (close to the average degree $\langle k \rangle$). The tail (high k region) of the degree distribution $P(k)$ decreases exponentially, which indicates that nodes that significantly deviate from the average are extremely rare. The clustering coefficient is independent of a node's degree, so $C(k)$ appears as a horizontal line if plotted as a function of k (C). The mean path length is proportional to the logarithm of the network size, $\langle l \rangle \sim \log(n)$, which indicates that it is characterized by the small-world property. (From Ref. [12].)

5.3 Random Networks

Random networks were invented by Erdös and Renyi [107–109] and are based on the principle that the probability p that there is an edge between any pair of vertices is distributed uniformly (Figure 14). The connectivity distribution $P(k)$ of a random network follows a Poisson distribution that peaks at the average vertex degree $\langle k \rangle$ (see Figure 14). Nodes that have a significantly lower or higher degree than $\langle k \rangle$ are absent or very rare. Moreover, random graphs have a relatively short average path length $\langle l \rangle$ which is proportional to the logarithm of the network size $\langle l \rangle \sim \log(n)$. The clustering coefficient for a random graph is expected to be constant with $\langle C \rangle = p$ (Figure 14). A detailed review of random graphs can be found in Bollobás [16].

5.4 Small-world Networks

Although complex networks do not look small at the first glance, there are many ways to travel from one vertex to another using a path of a few vertices only. This circumstance is also popularly known as "six degrees of separation" which goes back to a study by psychologist Stanley Milgram in 1967 [93]. Milgram concluded that two randomly chosen people are socially connected by an average of only five other people (resulting in six "degrees" between them). Such *small-world networks* have a short average path length

$\langle l \rangle$, a relatively large clustering coefficient $\langle C \rangle$, which is independent of the network size n, and a connectivity distribution $P(k)$, which is similar to that of a random graph. In a small-world network local perturbations can reach the whole network very quickly.

5.5 Scale-free Networks

If the connectivity of a network follows a power-law distribution, $P(k) \sim k^{-\gamma}$, the network is called scale-free. This relationship can also be seen as a straight line on a log-log plot since the above equation is equal to $\log(P(k)) \sim -y \cdot \log(k)$ (Figure 15B). Therefore, unlike that of a random graph, most vertices have only a few edges whereas a few vertices have a large number of edges, so called hubs (see light vertices in Figure 15A). However, they are also highly vulnerable to perturbations of highly connected nodes: when such hubs are removed the diameter increases rapidly and the network breaks into many isolated fragments [2]. By contrast, scale-free networks have a high degree of robustness against *random* errors.

If a network is scale-free, it is also a small-world. Scale-free networks with a degree exponent y between 2 and 3 are ultra-small with an average path length $\langle l \rangle \sim \log(\log(n))$ which is significantly shorter than $\langle l \rangle$ of a random graph which is proportionate to $\log(n)$, indicating that a heterogeneous scale-free topology is more efficient in bringing nodes close together than the homogeneous random graph topology [22, 23]. The clustering coefficient $\langle C \rangle$ of a scale-free network is about 5 times bigger than that of a random graph [1].

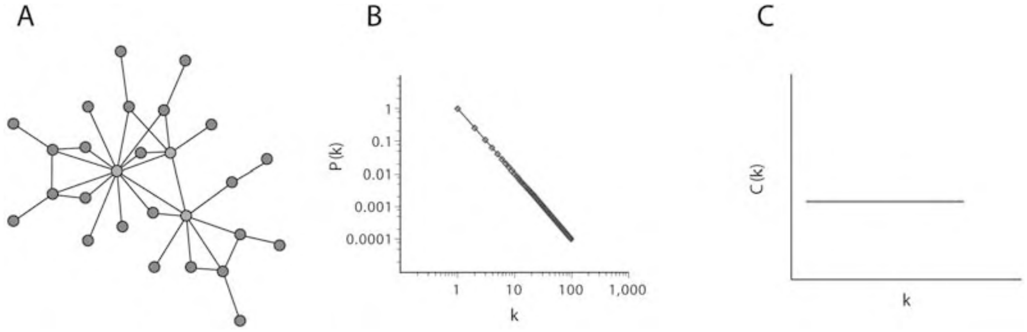


Figure 15 Scale-free network. (A) Small network with hubs (light). (B) Connectivity distribution of the network shown in (A). (C) Distribution of the clustering coefficient as a function of the vertex degree. For a large scale-free network see Figure 11 (From Ref. [12].)

5.6 Connectivity Distributions of Protein–Protein Interaction Networks

Several authors demonstrated that the connectivity distribution of the *Saccharomyces cerevisiae* protein–protein interaction network follows a power-law distribution [12]. They also determined that the interaction map of *H. pylori* [105] shows a heterogeneous scale-free network topology with a few highly connected and numerous less connected proteins. Furthermore, it has been shown that the interaction networks of *Drosophila melanogaster* [45] and *C. elegans* [85] proteins exhibit a distinct scale-free behavior.

This demonstrates that although the protein–protein interaction data sets have been derived from different sources and methods, the emergence of the scale-free property appears to be a robust feature [145]. However, several authors have argued that the scale-free nature of protein interaction networks may be an artifact of sampling and that these networks may be equally well described by other models [53, 124].

5.7 Error Tolerance and Attack Vulnerability

An important consequence of the scale-free connectivity distribution is the network's tolerance to random errors, coupled with fragility against the removal of the most connected nodes [2]. In fact, Jeong and coworkers [67] showed that highly connected proteins are 3 times more likely to be essential than proteins with only a small number of links to other proteins. In other words, when highly connected proteins are deleted the deletion is likely to be lethal. Similarly, Przulj and coworkers [104] demonstrated that nonessential proteins have a degree that is half that of essential proteins, whose deletion causes lethality. Interestingly, lethal mutations were not only found in highly connected proteins but also in proteins whose removal caused a disruption in network structure – removal of essential proteins disconnected one part of the network from the other. The importance of hubs is further validated by their evolutionary conservation. Fraser and coworkers demonstrated that highly interacting *S. cerevisiae* proteins have a smaller evolutionary distance to their orthologs in *C. elegans* than less-connected proteins [38], and Krylov and coworkers showed that yeast hubs are more likely to have orthologs proteins in higher organisms [78] (see also Sections 8.1 and 8.2).

Many cellular networks are fairly tolerant to random perturbations such as mutations, but they collapse when hubs are disrupted (by mutations or drugs). Network topology will also have an impact on drug development. For instance, highly connected proteins of pathogens may be suitable targets for an antibiotic therapy, whereas proteins which are less interconnected may be more appropriate targets for a highly specific drug in humans. A small degree is favored in human drug targets because it is more likely to avoid side effects (see also Chapters 18, 19 and 36).

5.8 Modules and Motifs in Networks

Another topological feature which reflects biological behavior is the hierarchical organization of scale-free networks [106]. Biological functions in a cell are organized in functional modules. Each module contains a group of physically or functionally linked molecules that work together to achieve a distinct function. For example, cells produce ATP via a set of modules, such as the glycolytic pathway, the Krebs cycle and the protein complexes involved in oxidative phosphorylation.

In networks, subgraphs (subsets of interconnected vertices) form triangles, squares, pentagons, etc. [94, 119]. Interestingly, some subgraphs, which are known as motifs, are more common in real networks than in randomized versions of the same network [94]. For example, triangle motifs are overrepresented in both transcription-regulatory and neural networks and four-node subgraphs are overrepresented in electric circuits. Each network is characterized by its own set of distinct motifs [63, 119].

A study on the evolutionary conservation of motif constituents within a network of yeast protein interactions uncovers a trend towards the preferential retention of highly cohesive motifs [144] (Figure 16).

Furthermore, empirical observations indicate that specific motif types aggregate to form large motif clusters. A number of algorithms have been developed to identify such modules (complexes or pathways) using either the network's topology [120] or combining it with functional genomics data [8, 65, 128]. Spirin and Mirny developed an algorithm that was able to recover many previously known modules such as the anaphase-promoting complex and the yeast pheromone response pathway [120] (Figure 17). The systematic identification of these modules provides essential knowledge linking proteome dynamics to cellular function and phenotype.

5.9 Comparing Protein Interaction Networks: Pathblast

Kelley and coworkers took the concept of interologs one step further by not only finding homologous interactions, but homologous pathways, here defined as paths of proteins that are connected by protein–protein interactions [71, 72] (Figure 18).

Pathblast can be used to compare, i.e. align, whole interaction networks. By using pathways even protein pairs with very weak sequence similarities can be identified because their position in the network provides additional information for the unambiguous identification of homologies. Thus, Kelley

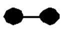
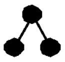
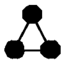
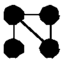
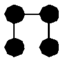
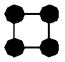
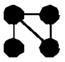
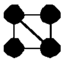
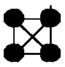
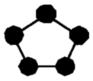
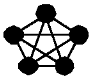
Motif	Number in yeast	Natural conservation	Random conservation	Conservation ratio
	9,266	13.67 %	4.63 %	2.94
	167,304	4.99 %	0.81 %	6.15
	3,846	20.51 %	1.01 %	20.28
	3,649,591	0.73 %	0.12 %	5.97
	1,763,891	2.64 %	0.18 %	14.67
	9,646	6.71 %	0.17 %	40.44
	164,075	7.67 %	0.17 %	45.56
	12,423	18.68 %	0.12 %	157.89
	2,339	32.53 %	0.08 %	422.78
	25,749	14.77 %	0.05 %	279.71
	1,433	47.24 %	0.02 %	2,256.67

Figure 16 The evolutionary conservation of motif constituents in yeast. Orthologs in five higher eukaryotes have been used to rate the conservation of yeast motifs (subgraphs of two to five nodes). The *number in yeast* is the absolute number found in a network of 3183 proteins taken from the DIP database [115]. The *natural conservation* rate is the fraction of yeast motifs that consists of proteins with orthologs in the five higher eukaryotes used. If the topology of motifs does not interfere with the conservation rate of its constituting proteins, a random ortholog distribution should give the same motif conservation rates as seen in the natural sample. The *random conservation* rate therefore represents the fraction of motifs that is fully conserved for the random ortholog distribution. The *conservation ratio* is the ratio between the natural and the random conservation ratios, indicating that all motifs are highly conserved. From Ref. [144].

and coworkers were able to compare protein interaction maps of yeast and *H. pylori*, i.e. between eukaryotes and prokaryotes. Completely unexpected similarities were found among the homologous pathways, e.g. membrane proteins in *Helicobacter* that are homologous to yeast proteins which are involved in transport through the nuclear pore (Figure 18).

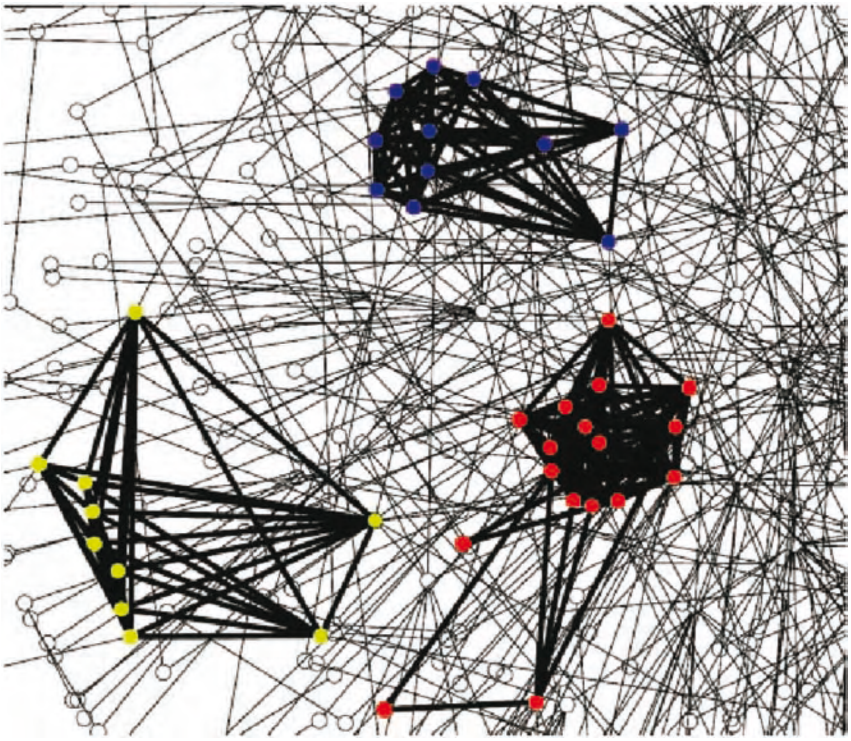


Figure 17 Computer algorithms can deduce molecular modules (protein complexes and pathways) directly from the topology of protein interaction networks. (From Ref. [120].)

6 Integrating Multiple Protein–Protein Interaction Evidence

Integrating protein interaction data means combining them with and relating them to other data (see also Chapter 42 for biological data integration).

There are two main reasons why integration of interaction data is useful:

- Integration aims at *collecting* all available data on certain proteins or groups of proteins, even whole genomes. Databases collect nonredundant information that is made available in machine-readable form so that computational analysis and relating data is possible. Such databases are also required for *visualization*, i.e. display of multidimensional information for human users (Figure 19; see also examples in Section 10).
- Integration of multiple data sets improves *annotation*, function *prediction*, *validation* and *experimental design* significantly. High-throughput data is often 1-D, i.e. a certain study may collect only information on protein interactions. In order to interpret this information it is necessary to add

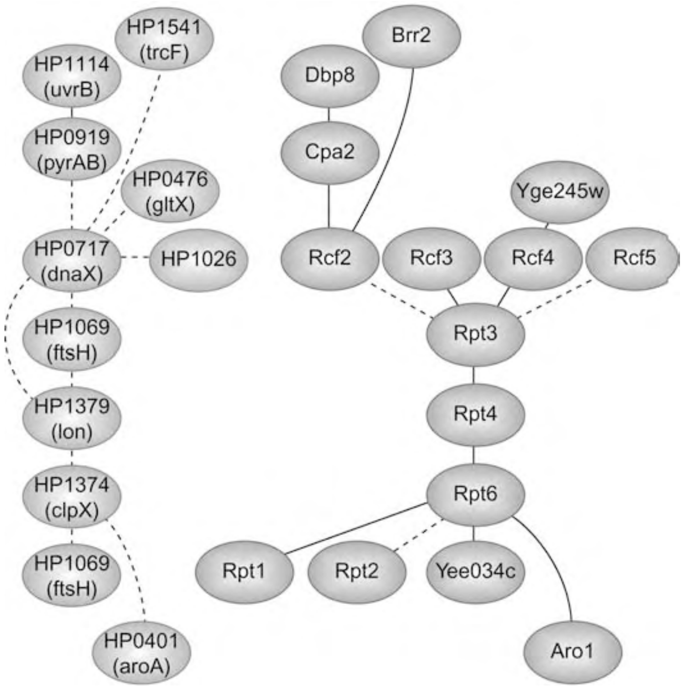


Figure 18 An example from Pathblast [71, 72]. The protein–protein interaction networks of *H. pylori* (left network) and *S. cerevisiae* (right network) were globally aligned to reveal conserved network regions. Proteins with above-threshold sequence similarity are placed on the same row of the pathway alignment (e.g. HP1114 and Dbp8). Direct protein interactions appear as solid links, and gaps or indirect interactions are dotted (From Ref. [71].)

information about expression, protein structure, localization, etc. Even previously uncharacterized proteins can be annotated fairly well from high-throughput data if all information is compiled and related [43, 129].

High-throughput data is also 1-D in the sense that it rarely provides information on the temporal and spatial dynamics and regulation of protein function. For example, in the endoplasmic reticulum so-called SREBP proteins form a complex with another membrane-embedded protein called SCAP, which escorts the SREBPs to the Golgi apparatus. Here, the SREBPs are sequentially cleaved by two Golgi-specific proteases. This releases a soluble fragment that travels to the nucleus where it interacts with other transcription factors to regulate its target genes (which are required for cholesterol synthesis). When cholesterol levels are low, SCAP escorts SREBPs to the Golgi, where processing takes place. When cholesterol levels are high, SCAP retains SREBP in the endoplasmic reticulum, processing is prevented and cholesterol synthesis is curtailed [96]. High-throughput studies (HTS) that analyze pro-

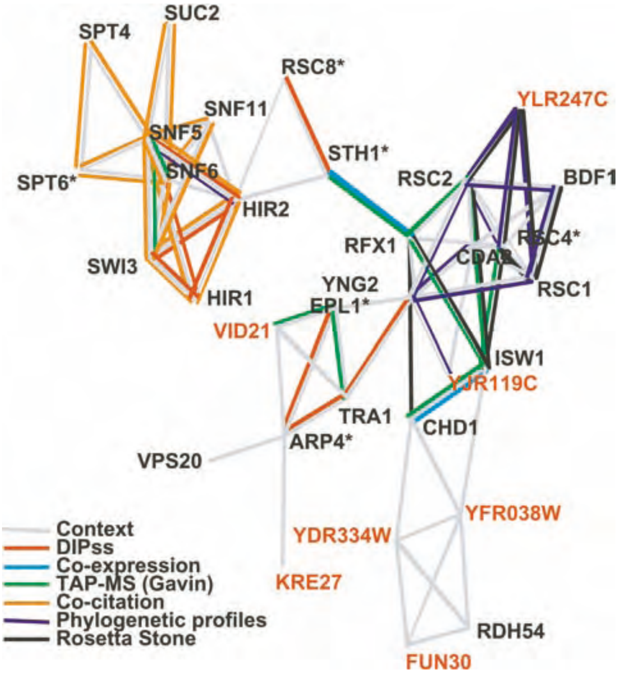


Figure 19 Integration of information. Shown are proteins involved in yeast chromatin remodeling, with several uncharacterized genes (red labels). This network was visualized with LGL (From Ref. [81], see Table 4).

tein interactions may only find nuclear interaction partners of SREBPs, HTS localization studies may only find their ER localization and expression studies may find that they are expressed at low levels – unless these assays are carried out under various cholesterol conditions.

Thus, high-throughput studies data will rarely give complete answers to biological problems. However, such data usually suggest further experiments and thus provide *hypotheses for hypothesis-driven research*. Protein interaction networks are particular informative because they often link proteins to other proteins whose function may suggest certain regulatory mechanisms. Even second-level protein interactions may be helpful, that is interactions of direct interactors.

Many studies integrated protein interaction data with other sources. Here we can present only a few examples:

- Lee and coworkers [81] reconstructed an extensive, high-quality functional gene network for yeast, consisting of 4681 (around 81%) of the known yeast genes linked by around 34 000 probabilistic linkages including expression data, gene linkage, phylogenetic profiles, cocitation and protein interaction data (Figure 19). The integrated linkages distinguish true- from false-positive interactions in earlier data sets; new interactions emerge from gene network contexts, as shown for genes in chromatin modification and ribosome biogenesis.

- Jansen and coworkers [65] used expression profiles, interaction data, and essentiality and localization information in order to predict protein complexes in yeast. Each of these data sources individually contains some weakly predictive information with respect to protein complexes, but Jansen and coworkers show how this prediction can be improved by combining all of them.

6.1 Protein Interactions and Gene Expression Data

Protein interactions and gene expression data have been correlated by a number of studies [15, 42, 50, 51, 128]. Although most of these studies found a correlation between interacting proteins and their gene expression levels, the correlation was not very strong. In fact, a few studies did not find a significant correlation or only did so when filtered data set were used [24,64]. Obviously, it is not necessary that expression levels are correlated in order to allow proteins to interact. See also Section 3.3.

6.2 Integration for Predicting Protein Function

Kemmeren and coworkers [75] analyzed 3.7 million yeast gene relationships (protein interactions, coexpression, colocalization, phenotypes, GO annotation) to predict the functions of yeast proteins or to improve their annotation. Integrated data were also used to cross-validate different interaction data sets which have different quality, different properties, etc. [74, 76]. See also Chapter 35 for further information on integrating heterogenous information to predict protein function.

7 Predicting Protein Functions from Protein Networks

Annotating uncharacterized proteins is still one of the most challenging problems of the post-genomic era [97]. However, assigning new functions to previously characterized proteins is equally important as hardly ever everything is known about a certain protein. Predicting protein function from sequence is discussed in Chapter 30. Inference of function from genomic context is described in Chapter 32 and predictions from structure in Chapter 33. Finally, Chapter 35 deals with information integration for protein function prediction.

We refer the reader to these chapters and only briefly discuss the role of protein interaction networks for protein function prediction.

Detecting a physical interaction between proteins is considered as one of the most powerful approaches for inferring the function of uncharacterized proteins [97].

Samanta and Liang [116] presented a network-based statistical algorithm that overcomes the presence of false positives in many interaction data sets and predicted functions of unannotated proteins from large-scale interaction data. Their algorithm uses the insight that if two proteins share a significantly larger number of common interaction partners than random, they have close functional associations. Analysis of publicly available data from *S. cerevisiae* revealed more than 2800 reliable functional associations, 29% of which involve at least one unannotated protein. By further analyzing these associations, Samanta and Liang derived functional predictions for 81 unannotated proteins with high certainty.

Vazquez and coworkers [135] presented an improved method for the assignment of protein functions based on global connectivity patterns of a protein network. In contrast to a simple majority rule [117], Vazquez and coworkers also used second-degree functional information, that is not only stemming from direct interactors but also from interactors of interactors.

Letovsky and Kasif [83] developed yet another method based on a probabilistic analysis of graph neighborhoods in a protein–protein interaction network to predict functions. The method exploits the fact that graph neighbors are more likely to share functions than nodes which are not neighbors. A binomial model of local neighbor function labeling probability was combined with a Markov random field propagation algorithm to assign function probabilities for proteins in the network.

8 Evolution of Protein–Protein Interactions

The evolution of protein networks can be studied from two different perspectives: (i) the network perspective where the loss and gain of interactions are counted for individual proteins, pathways or whole networks, and (ii) the protein sequence perspective, where the constraint of interactions on the rate of protein sequence evolution is measured. The former aspect is difficult to study because we do not have sufficient interaction data for different proteomes. Thus, most published studies concentrate on the sequence level.

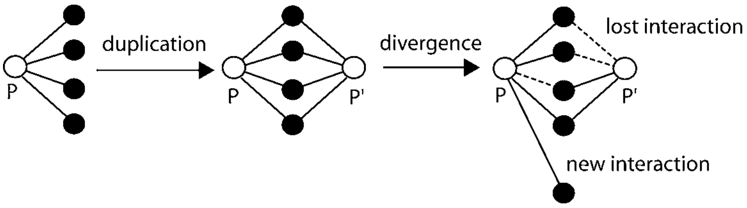


Figure 20 The effect of gene duplications on gene products that interact with proteins. Shortly after a gene duplication, the products P and P′ of the duplicate genes will interact with the same proteins. Eventually, some or all of the common interactions will be lost, and new interactions may be gained by either protein. In the rightmost panel, protein P has lost one interaction (dotted line) and gained a new interaction partner, whereas protein P′ has lost two interactions. If the number of common interaction partners is taken as a measure of functional overlap, then one of the functions of P is also covered by P′ and *vice versa*. (Redrawn after Ref. [138].)

8.1 The Network Level

Wagner [138] tried to estimate how fast the interaction network of yeast evolves, using gene duplicates (i.e. paralogs) in the yeast genome and the interaction partners of each duplicate (Figure 20).

Wagner found that genes with duplicates appeared slightly more highly connected. The reason is unclear. This difference in degree was not significant if more closely related duplicates were considered, i.e. duplicates with $K_s < 1$, where K_s is the fraction of synonymous (silent) substitutions per silent site.

More interestingly, Wagner's analysis indicated that duplicate gene products generally do not retain common interaction partners long after duplication. Only 57% (4/7) of the most closely related duplicate gene pairs ($0 < K_s < 0.5$) for which both genes interact with other proteins share any protein interaction partners. For all 380 gene pairs with $K_s > 0.5$, the fraction of duplicate partners with shared interactions is below 20%. For $K_s > 1.5$, it dwindles to a value close to the expected number of shared interactions between two proteins chosen at random from within the network.

Strikingly, already for $0.5 < K_s < 1$, only 20% of duplicate gene pairs share an interaction partner. That is, if one applies this criterion of functional overlap, 80% of genes have no functional overlap with their duplicates approximately 100 million years after the duplication (assuming a mutation rate of 10^{-8} per year per nucleotide).

8.1.1 The Rates of Interaction Loss and Gain

There are 127 duplicate gene pairs with $K_s < 2$ where both duplicates engage in protein–protein interactions. Assuming that all of the diversification observed between these duplicates is due to lost interactions, Wagner arrived at a total estimate of 920 interactions immediately after duplication, 429 of which have been lost since.

New interactions evolved at a rate of 2.88×10^{-6} per protein pair per million years according to Wagner. This may seem small. However, if extrapolated to all 1.97×10^{7}possible pairwise interactions in the yeast proteome, one arrives at an estimate of 57 newly evolving interactions per million years. Even when restricting oneself to the 985 proteins known to interact with other proteins, one arrives at an estimate of $(2.88 \times 10^{-6})(4.84 \times 10^{5}) \sim 1.4$ newly evolved interactions per million years. The true value is likely to lie somewhere in between.

Based on the assumption that the divergence in protein interactions after gene duplication is largely due to interaction loss, Wagner estimated the lower bound of the rate at which interactions get lost to be 2.2×10^{-3} per interaction per million years. If a comparable rate holds for interactions between single-copy genes, Wagner [138] estimated that 50% of all interactions may get lost every 300 million years.

8.2 Sequence and Interaction Divergence in Proteins

Interacting proteins have complementary surfaces that have evolved in a way that optimizes their affinity for the particular biological process they are involved in. Such interaction surfaces contain about 20 amino acids per partner (on average), although in most cases not all of these amino acids are involved in the interaction. Each protein loses about 800 Å^2 of contact area with the solvent when it is binding to its partner. It is intuitively obvious that the evolution of surface residues is constrained by protein–protein interactions, at least when the interaction is important for the cell. However, since the contact area comprises only a small fraction of the whole protein sequence, we will see an effect on the overall sequence only when the protein has many interactions with many different surface sites. Unfortunately, for most protein interactions we do not know their interaction sites. Thus, for simplification, the problem can be reduced to the question of how sequence conservation and the number of interactions are related.

Teichmann [126] found that proteins not known to be involved in interactions have an average sequence identity of 38% (between homologs of budding and fission yeast), while this value is 46% for proteins in stable complexes. Proteins that have transient interactions are intermediate between the two, with an average sequence identity of 41%. Thus, highly connected proteins (such as proteins in complexes) do appear to be more highly conserved than proteins with fewer connections. Indeed, Teichmann found that proteins belonging to small, medium and large complexes have average identities of

42, 44 and 51%, respectively (complexes were binned into groups with 1–14, 14–66 and 66–239 subunits).

8.2.1 Protein Evolution Rate and Protein–Protein Interactions

Fraser and coworkers [38] investigated the relationship between the evolutionary rate of protein sequence and the number of interactions these proteins have.

To estimate the evolutionary rates of these proteins, Fraser and coworkers compared putatively orthologous sequences between yeast and the nematode *C. elegans*. For each pair of orthologs, these authors estimated the evolutionary distance (K) that separates the two sequences, where K_a is defined as the number of substitutions per amino acid site that have taken place since the fungi–animal split. There were 164 yeast proteins for which the number of interactors and a well-conserved ortholog in the nematode was available.

If such coevolution is indeed an important mode of change in proteins constrained by interactions, then interacting proteins should evolve at similar rates. Fraser and coworkers tested this prediction by examining all 411 protein interactions in which each protein had a putative ortholog in *C. elegans* and showed no significant sequence similarity with its interaction partner. For each interaction, Fraser and coworkers calculated ΔK, which is the difference between the evolutionary distances separating the yeast proteins from their respective orthologs in the nematode. They then averaged these differences across all 411 interactions to find the mean difference in evolutionary rate between interacting proteins, $\Delta K^* = 1.3$ substitutions per site. To assess the significance of this difference, Fraser and coworkers repeatedly permuted the list of 411 interactions 10 000 times into random protein pairs and calculated the mean difference in evolutionary rate between arbitrarily paired proteins: In all but 44 of the 10 000 permutations, the observed averages of $\Delta K^* < \Delta K$, indicating that interacting proteins evolve at rates significantly closer than is expected to occur by chance ($p = 0.0044$).

A protein's fitness effect F, estimated as the reduction in relative growth rate of the organism due to deleting the gene that encodes the protein, is positively correlated with that protein's number of interactors I [38].

In summary, proteins with more interactors appear to evolve more slowly not because they are more important to the organism, but because a greater proportion of the protein is directly involved in its function. Interacting proteins evolve at similar rates.

The results by Fraser and coworkers [38] were not generally accepted. Jordan and coworkers [69] found only a very weak negative correlation between the number of interactions and evolutionary rate of a protein. In contrast to Fraser and coworkers, who used comparisons between yeast and worm proteins, Jordan and coworkers compared both budding and fission yeast as well as *H. pylori* and *Campylobacter jejuni*, both of which allow more reliable inference of homologous proteins than comparisons between yeast and worm. In addition, when the proteins from yeast were assorted into discrete bins according to the number of interactions, Jordan and coworkers found that only 6.5% of the proteins with the greatest number of interactions evolved, on average, significantly slower than the rest of the proteins.

While the correlation found by Fraser and coworkers was confirmed by Jordan and coworkers, even though it was weak, the result was dependent on the data set used: when only the most conserved (above 40% sequence identity), and thus most reliably identified, pairs of orthologous proteins were considered, the slope of the linear trend line decreased and the statistical significance disappeared.

As a conclusion, only a small fraction of yeast proteins with the largest number of interactions (the hubs of the interaction network) tend to evolve more slowly than the bulk of the proteins.

8.2.2 **Phylogenetic Relationships between Families of Interacting Proteins**

Goh and coworkers [46] and Goh and Cohen [47] reasoned that a ligand–receptor pair should occupy related positions in their phylogenetic trees if they coevolve. Previous results have shown that for ligand–receptor pairs that are part of large protein families, the correlation between their phylogenetic distance matrices is significantly greater than for unrelated protein families. An example of coevolving families is shown in Figure 21.

Following this logic, insulin sequences from different species should have a similar phylogenetic tree as the insulin receptor, because they coevolve.

As a consequence, the binding specificity of an uncharacterized protein may be inferred by comparing its phylogenetic tree to the trees of potential interaction partners. Results of the analysis show that binding partners can be quantitatively identified for proteins in diverse homologous protein families with approximately 58–100% sensitivity (predicted true binding pairs/all true binding pairs) and 82–100% specificity [1 – (predicted false binding pairs/all false binding pairs)] for a correlation score above 0.8 [47]. Similar results were obtained by Pazos and Valencia [101].

Hahn and coworkers [52] searched the genome of *S. cerevisiae* for the nearest paralog (if any) of each gene in the yeast protein-interaction network (that is an intragenome search) and used the ratio K_a/K_s, to measure selective constraint (K_a/K_s is the rate of amino acid replacement substitutions for each pair of orthologs divided by the rate of silent substitutions [77]). Furthermore, orthologous genes in the genomes of *Schizosaccharomyces pombe* and *Saccharomyces paradoxus* were identified.

Using the identified paralogs and orthologs, Hahn and coworkers calculated the correlation between evolutionary distance and the degree of protein

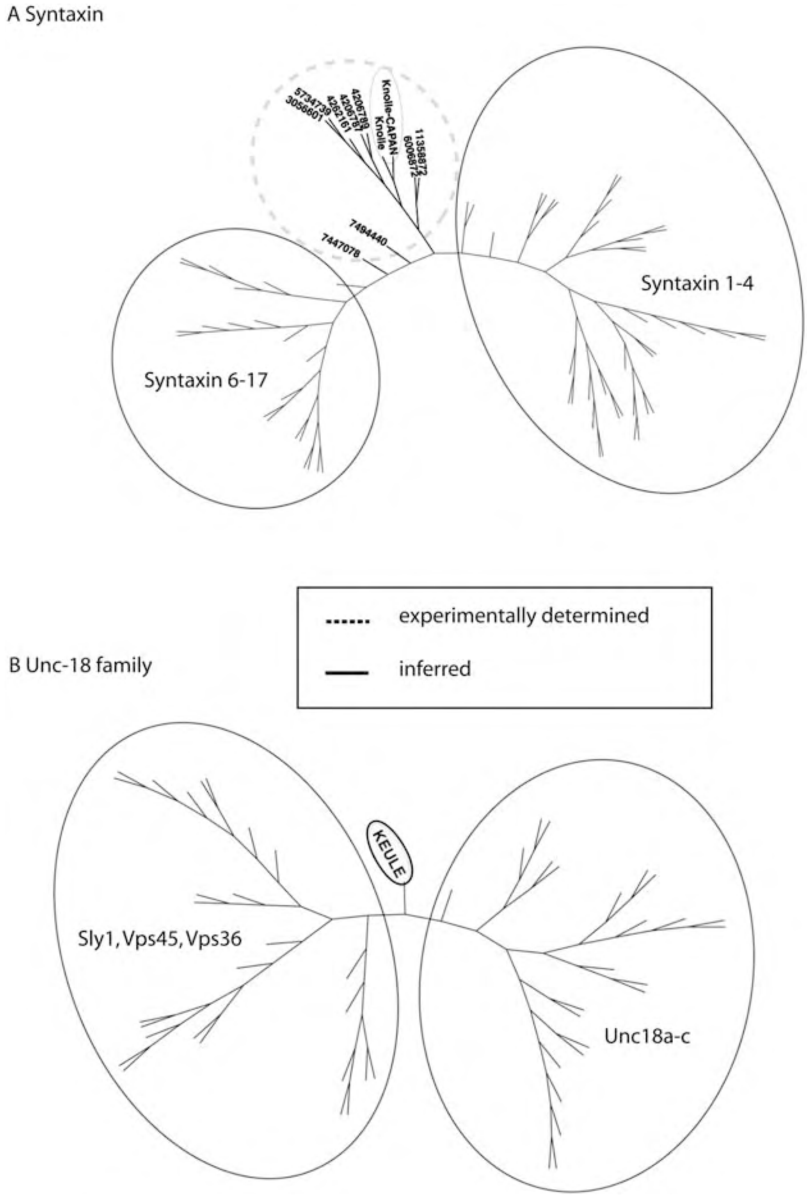


Figure 21 Coevolutionary analysis reveals insights into protein–protein interactions. Phylogenetic trees of the (a) syntaxin family and the (b) Unc-18 (Sec1) family. Members of the syntaxin family interact with members of the Unc-18 family and each family shows a similar tree structure. The area encompassed by the dotted circle indicates the search space for a potential KEULE binding partner. The filled circle outlines the known binding partner, Knolle, and an ortholog from *Capsicum anuum* (Knolle-CAPAN). (From Ref. [47].)

connectivity (D). For the *S. cerevisiae* paralogs and orthologs in *S. paradoxus*, they also calculated the correlation coefficients (both Pearson and Spearman) between K_a/K_s for the closest paralog and D using only unsaturated duplicate pairs with $K_s < 3$.

The study showed that there is a weakly significant Pearson's correlation between protein degree D and K_a/K_s, but no significant Spearman's correlation (Pearson's $r = -0.187$, $p = 0.047$; Spearman's $s = -0.151$, $p = 0.12$). There was a weak but highly significant correlation between D and the selective constraint (K_a/K_s) experienced by a gene.

Analyzing 1175 gene pairs from *S. cerevisiae* and *S. pombe* with this data set, Hahn and coworkers indeed found a correlation similar in magnitude to that obtained by Fraser and coauthors.

Hahn and coworkers were thus able to explain the discrepancy in results between Jordan and coauthors [69] and Fraser and coauthors [21,22] by using slightly different protein-interaction data sets. The reference taxon used by both of these groups, *S. pombe*, is less than ideal because it is only a distant relative of *S*. cerevisiae, with a most recent common ancestor 0.3–1.3 billion years ago. Thus, the analysis was repeated with the much more closely related *S. paradoxus* as the outgroup.

The relationship between the number of interacting partners and evolutionary constraint is highly dependent on a gene's function. Genes involved in metabolism, transport and the cytoskeleton show no significant relationship between D and K_a/K_s (always above 0.05). However, genes involved in the cell cycle and transcriptional processes show a significant, although weak, effect.

Pagel and coworkers [99] took a slightly different approach to investigate the relationship between interactions and evolution. They showed that interacting proteins in yeast have a much higher chance of possessing orthologs in other fungal genomes than randomly selected *S. cerevisiae* proteins. The number of species with orthologs was used here as a measure of conservation. This finding is compatible with the notion that highly connected nodes of the protein–protein interactions network are essential for survival.

Wuchty [143] also confirmed that there is a strong propensity of essential, highly connected proteins to be evolutionarily conserved, but he also found that this trend does not have an equivalent for nonessential proteins. Note the limitation to essential genes: only this subset of proteins shows correlation, not the complete set of interacting proteins. In order to guarantee balanced sampling for all connectivity (i.e. "k") values, Wuchty [143] used logarithmic binning of the k-axis, a procedure that corrects for the skewed nature of the scale-free distribution. In summary, Wuchty's results clearly indicate that highly connected proteins are far more likely to be essential and (simultaneously) conserved as orthologs in higher eukaryotes than are their less-connected counterparts.

Related studies have been published by Hurst and Smith [60], Hirsh and Fraser [57], Jordan and coworkers [68], and Fraser and coworkers [39].

8.3 Structural Aspects of Conserved Interactions

In addition to docking interacting proteins whose individual structure is known, one can study how interactions evolve on the structural level. For example, Aloy and coworkers [4] approached the problem by collecting all instances of the same domain pairs interacting in different complexes (from the SCOP database [7]), and then compared them with a simple geometric measure (interaction RMSD). When plotted against sequence similarity they found that close homologs (above 40% sequence identity) almost invariably interact the same way.

Aloy and coworkers defined contacting domains as those from the same Protein Data Bank (PDB) entry that had at least 10 C_α–C_α contacts smaller than 8 Å. The difference between domain orientations was described by the purely geometric interaction measure iRMSD. It does not require an interaction surface, interacting residues or residue equivalences to be assigned, and is thus readily applicable to remote similarities in sequence and over a wide range of differences in domain orientation. For identical domain interactions the expected iRMSD is 0, with values increasing with differences in domain orientations.

For some domain–domain interactions, interactions are preserved even at very low sequence identities, whereas for others the situation is reversed. For example, if one considers PID $< 20\%$ for the P-loop ATPase superfamily interacting with the ubiquitin-like superfamily, all four studied interactions are similar (iRMSD < 7 Å). In contrast, only two of the eight interactions between the P-loop ATPases and PH domains with PID $< 20\%$ have iRMSD < 10 Å, with the others showing great differences, their iRMSD being as high as 18 Å with clearly different binding surfaces.

This shows that interactions between weakly conserved proteins cannot always be predicted reliably (Figure 22). Additional experimental information may be required to construct realistic models in such cases.

9 Databases and Other Information Sources

Databases are critical for protein network analysis. A number of databases have been established over the past years. However, it is foreseeable that not all databases will continue to receive funding and thus we expect that

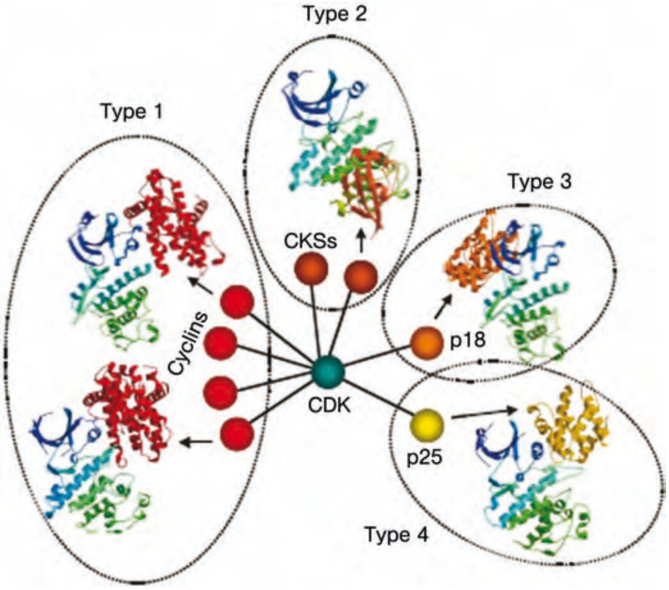


Figure 22 Homologous interactions are not always predictable. The same protein [here, cyclin-dependent kinase (CDK)] can have many different modes of interaction. This complicates both the prediction of homologous interactions and the relationship between connectivity and evolutionary constraint. See text for details. (From Ref. [5].)

some of theses databases will merge, while others may cease to exist. A list of databases is provided in Table 3.

10 Analysis and Visualization Tools

It is impossible to describe all available tools for protein network analysis and visualization in this chapter. In addition, new tools are constantly published and older tools improved or discontinued. We have collected some fairly widespread tools and their main features in Tables 4 and 5. Figure 23 shows a few screenshots from some of the packages we have used. We recommend visiting the websites of the academic or commercial suppliers for further information. New tools are regularly published in specialist journals.

11 Outlook/Perspectives

This chapter provides only a snapshot of a rapidly developing field in which many challenges remain. An increasing flood of data requires dynamic in-

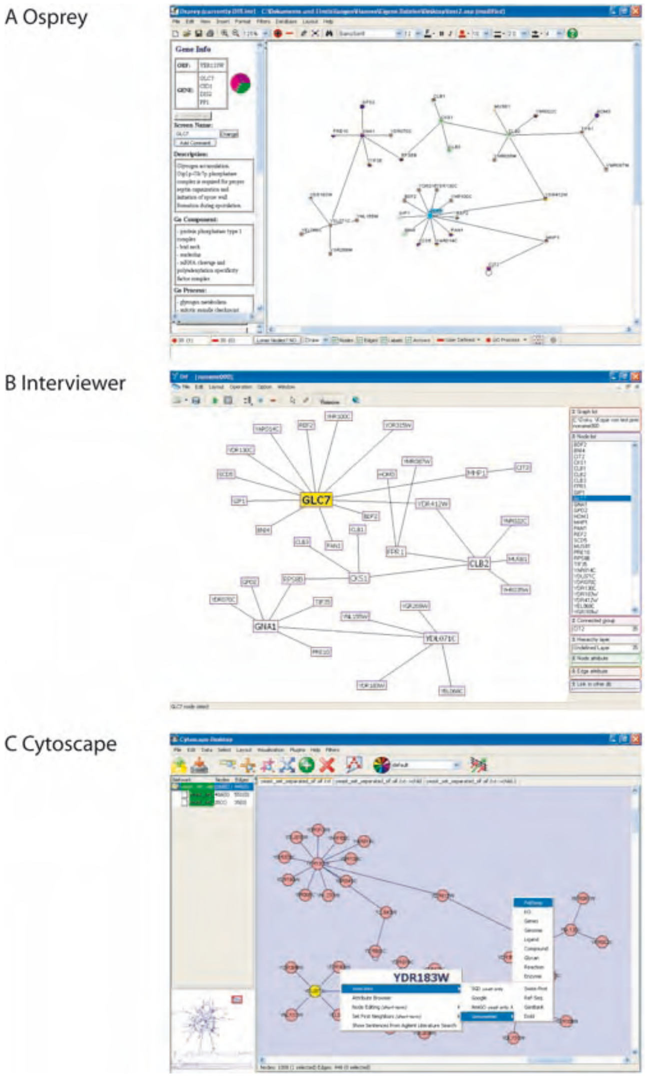


Figure 23 Protein–protein interaction network visualization tools. (A) Osprey 1.20 [18]. (B) WebInterViewer [54]. (C) Cytoscape 2.1 [118]. For details see Tables 4 and 5.

tegration into existing databases and constant re-evaluation of predictions and models. Integration of various data sources will be critical for biological interpretation. Some experimental approaches such as structural genomics are only now entering production levels and have barely been exploited for interactome analysis. Finally, comparative interactomics will only be possible with several completed interactomes. This is especially true for prokaryotic systems which are far more diverse than the few eukaryotes that can be

Table 4 Visualization and analysis tools

Name	Description	URL
Cytoscape[b]	visualization of protein–protein interactions networks and data integration	http://www.cytoscape.org
Graphviz	graph visualization software	http://www.graphviz.org
Bioverse Viewer	Java applet interface for visualizing interactomes	http://bioverse.compbio.washington.edu/viewer
WebInterViewer	visualization and analysis of protein–protein interactions	http://interviewer.inha.ac.kr
Osprey	network visualization system	http://biodata.mshri.on.ca/osprey/servlet/Index
ProViz	protein–protein interaction graph visualization tool	http://www.ebi.ac.uk/intact/doc/html/proviz/proviz.html
BioGraphNet	biological network visualization	http://llama.med.harvard.edu/BioGraphNet.html
Pajek	network visualization	http://vlado.fmf.uni-lj.si/pub/networks/pajek
Bind Viewer	network visualization	http://www.bind.ca
Graphbrowse	visualization system for interactive browsing of network diagrams	http://cvs.sourceforge.net/viewcvs.py/gmod/graphbrowse/graphbrowse_project_page.html?rev=1.3
Tulip library[b]	system dedicated to the visualization of huge graphs	http://www.tulip-software.org
JGraph library[b]	Java graph drawing and layout component	http://www.jgraph.com
yWorks[a]	Java, .NET libraries that provide algorithms to analyze, view, and draw graphs, diagrams and networks	http://www.yworks.com
Gravisto[b]	graph editor and visualization tool	http://www.gravisto.org
Wilmascope[b]	Java3D application which creates real time 3-D animations of dynamic graph structures.	http://www.wilmascope.org
GLuskap	software tool for displaying graphs in 3-D	http://www.cs.uleth.ca/∼vpak/gluskap
Walrus[b]	handles large directed graphs in 3-D space	http://www.caida.org/tools/visualization/walrus
LGL[b]	large graph layout; visualization tool dedicated to biological networks	http://bioinformatics.icmb.utexas.edu/lgl/
PathCalling	protein–protein interactions analysis and visualization	http://curatools.curagen.com/pathcalling_portal/index.htm
PimRider[a]	protein–protein interactions data and tool, *H. pylori*; free academic license available	http://pim.hybrigenics.com/pimriderext/common/
AiSee[a]	graph layout software	http://www.aisee.com/
Oreas[a]	C++, Java (over JNI), NET and COM. layout libraries and diagram editor	http://www.oreas.com/index_en.php

[a] Commercial.
[b] Open Source.

thoroughly studied experimentally. At the time of this writing not a single prokaryotic interactome was available (i.e. providing more than a small sample of interactions such as those published by Rain and coworkers and Malek and coworkers [87, 105]).

Table 5 A comparison of selected protein network visualization tools

	Cytoscape 2.1 [118]	Osprey 1.2.0 [19]	WebInterViewer [54]
URL	http://www.cytoscape.org	http://biodata.mshri.on.ca/osprey/servlet/Index	http://interviewer.inha.ac.kr/
Operating system(s)	all platforms for which a Java runtime environment (JRE) exists; Cytoscape 2.1 requires a 1.4.2 JRE	Windows XP, Red Hat Linux, Mac OS X	all platform for which a Java runtime environment (JRE) exists; WebInterViewer requires a 1.4.2 JRE
Source code	Open source under GNU LGPL	not freely available	not freely available
Import	Simple interaction file (.sif); a tab separated file with three columns (Node1 ↔ Interaction Type ↔ Node2)	custom Osprey network (.ocf .txt); a tab separated text file with up to seven columns (Node1 ↔ Node2 ↔ Alias1 ↔ Alias2 ↔ Experimental Method ↔ Source ↔ PubmedId)	a pair of interacting molecular names, separated by space or tab, in each line (.pnm) (Node1 ↔ Node2)
	plugin to import PSI-MI, level 1xml files (.xml) (other levels will be provided in the future)	GeneList (.gl .txt); a tab separated text file with one column containing gene names	a pair of interacting molecular indices, separated by tab, in each line (.pid)
		interactions from the GRID database via the Internet	a pair of molecular index and its name, separated by tab, in each line (.pid_label)
			an ordered list of molecular index and its position (x-, y- and z-coordinates) separated by tab, in each line (.pid_pos)
			data in xin format from DIP (.xin)
Network layout format	graph markup language (.gml)	Osprey files (.osp)	graph markup language (.gml)
Import node/edge attributes	node and edge attribute files containing node annotations or numerical edge values, e.g. confidence values or expression data	experimental method, source and PubmedId via custom Osprey network (.ocf .txt)	data on molecule, domain, and function in XML format (.xml)
Navigation	zoom in/out zoom selected region navigation panel with an a bird's eye view	zoom in/out	zoom in/out graph rotation
Layout	spring embedded layout circular organic hierarchic	auto relaxation circular concentric circles dual ring	F-D layout RSFDP
Adjustable visual styles	color, shape and size/ thickness of nodes and edges	color and size of nodes and edges	color, shape and size of nodes
	color gradient on numerical edge attributes	font type and size of node labels	node size proportional to its degree
	mapping colors/shapes onto specific interactiontypes	color nodes by GO process	font type and size of node labels
		color edges by experimental system or source	labeling nodes by their node degree
Network editing	drag and drop of network segments	drag and drop of network segments	drag and drop of nodes
	squiggle feature to mark up the network	change node name and add node comments	selection of connected groups
	hiding nodes and edges	hiding nodes and edges	hiding nodes and edges
	aligning and rotating of groups of nodes	remove loner nodes	
Multiple networks	multiple networks can be loaded at a time	one network can be loaded at a time	multiple networks can be loaded at a time
	new network from selected nodes and edges	superimposing of another network	new network from selected nodes and edges
			find common nodes within two or more networks
			union with two or more networks

Table 5 (continued)

	Cytoscape 2.1 [118]	Osprey 1.2.0 [19]	WebInterViewer [54]
Global filters	filter node labels using * and ? as wildcards filter nodes, based on the edges that they are connected to >, = and < filtering operations on numerical edge attributes filter nodes based on the number of edges to other nodes combine filters together using AND, OR and XOR operators	filter nodes based on their source or experimental method filter nodes based on their GO process or processes filter nodes based on the number of edges connected to them filter nodes until the remaining nodes have a certain number of interactions	
Information through annotation server	This feature is currently only for *S. cerevisiae*: SGD Google AmiGO GenomeNet (Pathway, KO, Genes, Genome, Ligand, Compound, Glycan, Enzyme, Reaction, Swiss-Prot, GenBank, Ref-Seq, EMBL)	available for all organisms in BioGRID database ORF names and gene/alias names Go Component, Go Process and Go Function experimental system/sources PubMed ID: a link to any PubMed listing on a given interaction	connection to dataserver failed
Image export	encapsulated postscript (.eps, .epi, .epsi, .epsf) graphics interchange format (.gif) Joint Photographers Expert Group format (.jpg, .jpeg) MacroMedia flash file format (.swf) portable document format (.pdf) portable network graphics format (.png) postscript (.ps) RAW image (.raw) scaleable vector graphics (.svgz, .svg)	scaleable vector graphic (.svg) Joint Photographers Expert Group format (.jpg, .jpeg) portable network graphics format (.png) matrix (.txt)	Joint Photographers Expert Group format (.jpg, .jpeg) portable network graphics format (.png)
Advantages	can handle large networks fast and good layouting with high clarity can handle large networks high graphical network solution integration of expression data highly adjustable (plugin framework)	GO annotation and clustering direct import of interactions from BioGRID	can handle large networks easy navigation fast layouting 3-D navigation easy selection of connected groups
Disadvantages		slow layouting, low-resolution graphics	no filters, connection to dataserver failed
Documentation	detailed	detailed documentation	scarce

With increasing amounts of experimental data, analysis and visualization tools will be increasingly important. In particular, such tools need to be user-friendly enough to be used by experimental biologists. Eventually this will hopefully allow us to do “real” systems biology, i.e. representing and simulating biological systems in all their 4-D complexity *in silico* [125, 131].

References

1 ALBERT, R. AND A.L. BARABASI. 2002. Statistical mechanics of complex networks. Rev. Mod. Phys. **74**: 47–97.

2 ALBERT, R., H. JEONG AND A. L. BARABASI. 2000. Error and attack tolerance of complex networks. Nature **406**: 378–82.

3 ALBERTS, B., A. JOHNSON, J. LEWIS, M. RAFF, K. ROBERTS AND P. WALTER. 2002. *Molecular Biology of the Cell*, 4th edn. Garland, New York, NY.

4 ALOY, P., H. CEULEMANS, A. STARK AND R. B. RUSSELL. 2003. The relationship between sequence and interaction divergence in proteins. J. Mol. Biol. **332**: 989–98.

5 ALOY, P. AND R. B. RUSSELL. 2004. Ten thousand interactions for the molecular biologist. Nat. Biotechnol. **22**: 1317–21.

6 ALOY, P. AND R. B. RUSSELL. 2002. The third dimension for protein interactions and complexes. Trends Biochem. Sci. **27**: 633–8.

7 ANDREEVA, A., D. HOWORTH, S. E. BRENNER, T. J. HUBBARD, C. CHOTHIA AND A. G. MURZIN. 2004. SCOP database in 2004: refinements integrate structure and sequence family data. Nucleic Acids Res. **32**: D226–9.

8 BADER, G. D. AND C. W. HOGUE. 2003. An automated method for finding molecular complexes in large protein interaction networks. BMC Bioinformatics **4**: 2.

9 BADER, J. S., A. CHAUDHURI, J. M. ROTHBERG AND J. CHANT. 2004. Gaining confidence in high-throughput protein interaction networks. Nat. Biotechnol. **22**: 78–85.

10 BARABASI, A. L. AND R. ALBERT. 1999. Emergence of scaling in random networks. Science **286**: 509–12.

11 BARABASI, A. L. AND H. JEONG. 1999. Mean-field theory for scale-free random networks. Physica A **272**: 173–97.

12 BARABASI, A. L. AND Z. N. OLTVAI. 2004. Network biology: understanding the cell's functional organization. Nat. Rev. Genet. **5**: 101–13.

13 BARTEL, P. AND S. FIELDS. 1997. *The Yeast Two-hybrid System*. Oxford University Press, Oxford.

14 BARTEL, P. L., J. A. ROECKLEIN, D. SENGUPTA AND S. FIELDS. 1996. A protein linkage map of *Escherichia coli* bacteriophage T7. Nat. Genet. **12**: 72–7.

15 BHARDWAJ, N. AND H. LU. 2005. Correlation between gene expression profiles and protein–protein interactions within and across genomes. Bioinformatics **21**: 2730–8.

16 BOLLOBÁS, B. 1985. *Random Graphs*. Academic Press, London.

17 BOUWMEESTER, T., A. BAUCH, H. RUFFNER, et al. 2004. A physical and functional map of the human TNF-alpha/NF-kappa B signal transduction pathway. Nat. Cell Biol. **6**: 97–105.

18 BREITKREUTZ, B. J., C. STARK AND M. TYERS. 2003. Osprey: a network visualization system. Genome Biol. **4**: R22.

19 BROWN, K. R. AND I. JURISICA. 2005. Online predicted human interaction database. Bioinformatics **21**: 2076–82.

20 BUNESCU, R., R. GE, R. J. KATE, E. M. MARCOTTE, R. J. MOONEY, A. K. RAMANI AND Y. W. WONG. 2005. Comparative experiments on learning information extractors for proteins and their interactions. Artif. Intell. Med. **33**: 139–55.

21 BUTLAND, G., J. M. PEREGRIN-ALVAREZ, J. LI, et al. 2005. Interaction network containing conserved and essential protein complexes in *Escherichia coli*. Nature **433**: 531–7.

22 CHUNG, F. AND L. LU. 2002. The average distances in random graphs with given expected degrees. Proc. Natl Acad. Sci. USA **99**: 15879–82.

23 COHEN, R. AND S. HAVLIN. 2003. Scale-free networks are ultrasmall. Phys. Rev. Lett **90**: 058701.

24 CORNELL, M., N. W. PATON AND S. G. OLIVER. 2004. A critical and integrated view of the yeast interactome. Comp. Funct. Genomics **5**: 382–402.

25 Cramer, P., D. A. Bushnell and R. D. Kornberg. 2001. Structural basis of transcription: RNA polymerase II at 2.8 angstrom resolution. Science **292**: 1863–76.

26 Dandekar, T., B. Snel, M. Huynen and P. Bork. 1998. Conservation of gene order: a fingerprint of proteins that physically interact. Trends Biochem. Sci. **23**: 324–8.

27 Date, S. V. and E. M. Marcotte. 2005. Protein function prediction using the Protein Link EXplorer (PLEX). Bioinformatics **21**: 2558–9.

28 Deane, C. M., L. Salwinski, I. Xenarios and D. Eisenberg. 2002. Protein interactions: two methods for assessment of the reliability of high throughput observations. Mol. Cell Proteomics **1**: 349–56.

29 Deng, M., S. Mehta, F. Sun and T. Chen. 2002. Inferring domain-domain interactions from protein–protein interactions. Genome Res. **12**: 1540–8.

30 Edwards, A. M., B. Kus, R. Jansen, D. Greenbaum, J. Greenblatt and M. Gerstein. 2002. Bridging structural biology and genomics: assessing protein interaction data with known complexes. Trends Genet. **18**: 529–36.

31 Enright, A. J., I. Iliopoulos, N. C. Kyrpides and C. A. Ouzounis. 1999. Protein interaction maps for complete genomes based on gene fusion events. Nature **402**: 86–90.

32 Eyre, T. A., F. Ducluzeau, T. P. Sneddon, S. Povey, E. A. Bruford and M. J. Lush. 2006. The HUGO Gene Nomenclature Database, 2006 updates. Nucleic Acids Res. **34**: D319–21.

33 Fayos, R., G. Melacini, M. G. Newlon, L. Burns, J. D. Scott and P. A. Jennings. 2003. Induction of flexibility through protein–protein interactions. J Biol. Chem **278**: 18581–7.

34 Fields, S. and O. Song. 1989. A novel genetic system to detect protein–protein interactions. Nature **340**: 245–6.

35 Figeys, D. 2003. Novel approaches to map protein interactions. Curr. Opin. Biotechnol. **14**: 119–25.

36 Flores, A., J. F. Briand, O. Gadal, et al. 1999. A protein–protein interaction map of yeast RNA polymerase III. Proc. Natl Acad. Sci. USA **96**: 7815–20.

37 Formstecher, E., S. Aresta, V. Collura, et al. 2005. Protein interaction mapping: a Drosophila case study. Genome Res. **15**: 376–84.

38 Fraser, H. B., A. E. Hirsh, L. M. Steinmetz, C. Scharfe and M. W. Feldman. 2002. Evolutionary rate in the protein interaction network. Science **296**: 750–2.

39 Fraser, H. B., D. P. Wall and A. E. Hirsh. 2003. A simple dependence between protein evolution rate and the number of protein–protein interactions. BMC Evol Biol. **3**: 11.

40 Fromont-Racine, M., J. C. Rain and P. Legrain. 1997. Toward a functional analysis of the yeast genome through exhaustive two-hybrid screens. Nat. Genet. **16**: 277–82.

41 Gavin, A. C., M. Bosche, R. Krause, et al. 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature **415**: 141–7.

42 Ge, H., Z. Liu, G. M. Church and M. Vidal. 2001. Correlation between transcriptome and interactome mapping data from *Saccharomyces cerevisiae*. Nat. Genet. **29**: 482–6.

43 Ge, H., A. J. Walhout and M. Vidal. 2003. Integrating "omic" information: a bridge between genomics and systems biology. Trends Genet. **19**: 551–60.

44 Gene Ontology Consortium. 2006. The Gene Ontology (GO) project in 2006. Nucleic Acids Res. **34**: D322–6.

45 Giot, L., J. S. Bader, C. Brouwer, et al. 2003. A protein interaction map of Drosophila melanogaster. Science **302**: 1727–36.

46 Goh, C. S., A. A. Bogan, M. Joachimiak, D. Walther and F. E. Cohen. 2000. Co-evolution of proteins with their interaction partners. J. Mol. Biol. **299**: 283–93.

47 Goh, C. S. and F. E. Cohen. 2002. Co-evolutionary analysis reveals insights into protein–protein interactions. J. Mol. Biol. **324**: 177–92.

48 Goh, C. S., D. Milburn and M. Gerstein. 2004. Conformational

changes associated with protein–protein interactions. Curr. Opin. Struct Biol. **14**: 104–9.

49 Goldberg, D. S. and F. P. Roth. 2003. Assessing experimentally derived interactions in a small world. Proc. Natl Acad. Sci. USA **100**: 4372–6.

50 Grigoriev, A. 2001. A relationship between gene expression and protein interactions on the proteome scale: analysis of the bacteriophage T7 and the yeast *Saccharomyces cerevisiae*. Nucleic Acids Res. **29**: 3513–9.

51 Hahn, A., J. Rahnenfuhrer, P. Talwar and T. Lengauer. 2005. Confirmation of human protein interaction data by human expression data. BMC Bioinformatics **6**: 112.

52 Hahn, M. W., G. C. Conant and A. Wagner. 2004. Molecular evolution in large genetic networks: does connectivity equal constraint? J. Mol. Evol. **58**: 203–11.

53 Han, J. D., D. Dupuy, N. Bertin, M. E. Cusick and M. Vidal. 2005. Effect of sampling on topology predictions of protein–protein interaction networks. Nat. Biotechnol. **23**: 839–44.

54 Han, K., B. H. Ju and H. Jung. 2004. WebInterViewer: visualizing and analyzing molecular interaction networks. Nucleic Acids Res. **32**: W89–95.

55 Hao, Y., X. Zhu, M. Huang and M. Li. 2005. Discovering patterns to extract protein–protein interactions from the literature: part II. Bioinformatics **21**: 3294–300.

56 Hermjakob, H., L. Montecchi-Palazzi, C. Lewington, et al. 2004. IntAct: an open source molecular interaction database. Nucleic Acids Res. **32**: D452–5.

57 Hirsh, A. E. and H. B. Fraser. 2001. Protein dispensability and rate of evolution. Nature **411**: 1046–9.

58 Ho, Y., A. Gruhler, A. Heilbut, et al. 2002. Systematic identification of protein complexes in *Saccharomyces cerevisiae* by mass spectrometry. Nature **415**: 180–3.

59 Huh, W. K., J. V. Falvo, L. C. Gerke, A. S. Carroll, R. W. Howson, J. S. Weissman and E. K. O'Shea. 2003. Global analysis of protein localization in budding yeast. Nature **425**: 686–91.

60 Hurst, L. D. and N. G. Smith. 1999. Do essential genes evolve slowly? Curr. Biol. **9**: 747–50.

61 Huynen, M., B. Snel, W. Lathe, 3rd and P. Bork. 2000. Predicting protein function by genomic context: quantitative evaluation and qualitative inferences. Genome Res. **10**: 1204–10.

62 Ito, T., T. Chiba, R. Ozawa, M. Yoshida, M. Hattori and Y. Sakaki. 2001. A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proc. Natl Acad. Sci. USA **98**: 4569–74.

63 Itzkovitz, S., R. Milo, N. Kashtan, G. Ziv and U. Alon. 2003. Subgraphs in random networks. Phys Rev. E **68**: 026127.

64 Jansen, R., D. Greenbaum and M. Gerstein. 2002. Relating whole-genome expression data with protein–protein interactions. Genome Res. **12**: 37–46.

65 Jansen, R., N. Lan, J. Qian and M. Gerstein. 2002. Integration of genomic datasets to predict protein complexes in yeast. J. Struct. Funct. Genomics **2**: 71–81.

66 Jenssen, T. K., A. Laegreid, J. Komorowski and E. Hovig. 2001. A literature network of human genes for high-throughput analysis of gene expression. Nat. Genet. **28**: 21–8.

67 Jeong, H., S. P. Mason, A. L. Barabasi and Z. N. Oltvai. 2001. Lethality and centrality in protein networks. Nature **411**: 41–2.

68 Jordan, I. K., I. B. Rogozin, Y. I. Wolf and E. V. Koonin. 2002. Essential genes are more evolutionarily conserved than are nonessential genes in bacteria. Genome Res. **12**: 962–8.

69 Jordan, I. K., Y. I. Wolf and E. V. Koonin. 2003. No simple dependence between protein evolution rate and the number of protein–protein interactions: only the most prolific interactors tend to evolve slowly. BMC Evol. Biol. **3**: 1.

70 Kanehisa, M., S. Goto, M. Hattori, et al. 2006. From genomics to chemical genomics: new developments in KEGG. Nucleic Acids Res. **34**: D354–7.

71 Kelley, B. P., R. Sharan, R. M. Karp, T. Sittler, D. E. Root, B. R.

STOCKWELL AND T. IDEKER. 2003. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. Proc. Natl Acad. Sci. USA **100**: 11394–9.

72 KELLEY, B. P., B. YUAN, F. LEWITTER, R. SHARAN, B. R. STOCKWELL AND T. IDEKER. 2004. PathBLAST: a tool for alignment of protein interaction networks. Nucleic Acids Res. **32**: W83–8.

73 KELLEY, R. AND T. IDEKER. 2005. Systematic interpretation of genetic interactions using protein networks. Nat. Biotechnol. **23**: 561–6.

74 KEMMEREN, P. AND F. C. HOLSTEGE. 2003. Integrating functional genomics data. Biochem. Soc. Trans. **31**: 1484–7.

75 KEMMEREN, P., T. T. KOCKELKORN, T. BIJMA, R. DONDERS AND F. C. HOLSTEGE. 2005. Predicting gene function through systematic analysis and quality assessment of high-throughput data. Bioinformatics **21**: 1644–52.

76 KEMMEREN, P., N. L. VAN BERKUM, J. VILO, T. BIJMA, R. DONDERS, A. BRAZMA AND F. C. HOLSTEGE. 2002. Protein interaction verification and functional annotation by integrated analysis of genome-scale data. Mol. Cell **9**: 1133–43.

77 KIMURA, M. 1977. Preponderance of synonymous changes as evidence for the neutral theory of molecular evolution. Nature **267**: 275–6.

78 KRYLOV, D. M., Y. I. WOLF, I. B. ROGOZIN AND E. V. KOONIN. 2003. Gene loss, protein sequence divergence, gene dispensability, expression level, and interactivity are correlated in eukaryotic evolution. Genome Res. **13**: 2229–35.

79 LABAER, J. 2003. Mining the literature and large datasets. Nat. Biotechnol. **21**: 976–7.

80 LACOUNT, D. J., M. VIGNALI, R. CHETTIER, et al. 2005. A protein interaction network of the malaria parasite *Plasmodium falciparum*. Nature **438**: 103–7.

81 LEE, I., S. V. DATE, A. T. ADAI AND E. M. MARCOTTE. 2004. A probabilistic functional network of yeast genes. Science **306**: 1555–8.

82 LEHNER, B. AND A. G. FRASER. 2004. A first-draft human protein-interaction map. Genome Biol. **5**: R63.

83 LETOVSKY, S. AND S. KASIF. 2003. Predicting protein function from protein/protein interaction data: a probabilistic approach. Bioinformatics **19** SUPPL **1**: i197–204.

84 LETUNIC, I., R. R. COPLEY, B. PILS, S. PINKERT, J. SCHULTZ AND P. BORK. 2006. SMART 5: domains in the context of genomes and networks. Nucleic Acids Res. **34**: D257–60.

85 LI, S., C. M. ARMSTRONG, N. BERTIN, et al. 2004. A map of the interactome network of the metazoan *C. elegans*. Science **303**: 540–3.

86 MAGLOTT, D., J. OSTELL, K. D. PRUITT AND T. TATUSOVA. 2005. Entrez Gene: gene-centered information at NCBI. Nucleic Acids Res. **33**: D54–8.

87 MALEK, J. A., J. M. WIERZBOWSKI, W. TAO, et al. 2004. Protein interaction mapping on a functional shotgun sequence of *Rickettsia sibirica*. Nucleic Acids Res. **32**: 1059–64.

88 MANN, M., R. C. HENDRICKSON AND A. PANDEY. 2001. Analysis of proteins and proteomes by mass spectrometry. Annu. Rev. Biochem. **70**: 437–73.

89 MARCOTTE, E. M., M. PELLEGRINI, H. L. NG, D. W. RICE, T. O. YEATES AND D. EISENBERG. 1999. Detecting protein function and protein–protein interactions from genome sequences. Science **285**: 751–3.

90 MARCOTTE, E. M., I. XENARIOS AND D. EISENBERG. 2001. Mining literature for protein–protein interactions. Bioinformatics **17**: 359–63.

91 MATTHEWS, L. R., P. VAGLIO, J. REBOUL, H. GE, B. P. DAVIS, J. GARRELS, S. VINCENT AND M. VIDAL. 2001. Identification of potential interaction networks using sequence-based searches for conserved protein–protein interactions or "interologs". Genome Res. **11**: 2120–6.

92 MEWES, H. W., D. FRISHMAN, K. F. MAYER, et al. 2006. MIPS: analysis and annotation of proteins from whole genomes in 2005. Nucleic Acids Res. **34**: D169–72.

93 Milgram, S. 1967. The small world problem. Psychol. Today **1**: 61–67.

94 Milo, R., S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii and U. Alon. 2002. Network motifs: simple building blocks of complex networks. Science **298**: 824–7.

95 Ng, S. K., Z. Zhang and S. H. Tan. 2003. Integrative approach for computationally inferring protein domain interactions. Bioinformatics **19**: 923–9.

96 Nohturfft, A. and R. Losick. 2002. Cell biology. Fats, flies, and palmitate. Science **296**: 857–8.

97 Okada, K., S. Kanaya and K. Asai. 2005. Accurate extraction of functional associations between proteins based on common interaction partners and common domains. Bioinformatics **21**: 2043–8.

98 Oyama, T., K. Kitano, K. Satou and T. Ito. 2002. Extraction of knowledge on protein–protein interaction by association rule discovery. Bioinformatics **18**: 705–14.

99 Pagel, P., H. W. Mewes and D. Frishman. 2004. Conservation of protein–protein interactions – lessons from ascomycota. Trends Genet. **20**: 72–6.

100 Pazos, F. and A. Valencia. 2002. *In silico* two-hybrid system for the selection of physically interacting protein pairs. Proteins **47**: 219–27.

101 Pazos, F. and A. Valencia. 2001. Similarity of phylogenetic trees as indicator of protein–protein interaction. Protein Eng. **14**: 609–14.

102 Pellegrini, M., E. M. Marcotte, M. J. Thompson, D. Eisenberg and T. O. Yeates. 1999. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. Proc. Natl Acad. Sci. USA **96**: 4285–8.

103 Phizicky, E. M. and S. Fields. 1995. Protein–protein interactions: methods for detection and analysis. Microbiol. Rev. **59**: 94–123.

104 Przulj, N., D. A. Wigle and I. Jurisica. 2004. Functional topology in a network of protein interactions. Bioinformatics **20**: 340–8.

105 Rain, J. C., L. Selig, H. De Reuse, et al. 2001. The protein–protein interaction map of *Helicobacter pylori*. Nature **409**: 211–215.

106 Ravasz, E., A. L. Somera, D. A. Mongru, Z. N. Oltvai and A. L. Barabasi. 2002. Hierarchical organization of modularity in metabolic networks. Science **297**: 1551–5.

107 Renyi, P. E. A. A. 1959. On random graphs. Pub. Math. **6**: 290–7.

108 Renyi, P. E. A. A. 1960. On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci. **5**: 17–61.

109 Renyi, P. E. A. A. 1961. On the strength of connectedness of a random graph. Acta Math. Sci. Hung. **12**: 261–7.

110 Rhodes, D. R., S. A. Tomlins, S. Varambally, et al. 2005. Probabilistic model of the human protein–protein interaction network. Nat. Biotechnol. **23**: 951–9.

111 Rual, J. F., K. Venkatesan, T. Hao et al. 2005. Towards a proteome-scale map of the human protein–protein interaction network. Nature **437**: 1173–8.

112 Saito, R., H. Suzuki and Y. Hayashizaki. 2003. Construction of reliable protein–protein interaction networks with a new interaction generality measure. Bioinformatics **19**: 756–63.

113 Saito, R., H. Suzuki and Y. Hayashizaki. 2002. Interaction generality, a measurement to assess the reliability of a protein–protein interaction. Nucleic Acids Res. **30**: 1163–8.

114 Salwinski, L. and D. Eisenberg. 2003. Computational methods of analysis of protein–protein interactions. Curr. Opin. Struct. Biol. **13**: 377–382.

115 Salwinski, L., C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie and D. Eisenberg. 2004. The Database of Interacting Proteins: 2004 update. Nucleic Acids Res. **32**: D449–451.

116 Samanta, M. P. and S. Liang. 2003. Predicting protein functions from redundancies in large-scale protein interaction networks. Proc. Natl Acad. Sci. USA **100**: 12579–83.

117 SCHWIKOWSKI, B., P. UETZ AND S. FIELDS. 2000. A network of protein–protein interactions in yeast. Nat. Biotechnol. **18**: 1257–61.

118 SHANNON, P., A. MARKIEL, O. OZIER, et al. 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. **13**: 2498–504.

119 SHEN-ORR, S. S., R. MILO, S. MANGAN AND U. ALON. 2002. Network motifs in the transcriptional regulation network of *Escherichia coli*. Nat. Genet. **31**: 64–8.

120 SPIRIN, V. AND L. A. MIRNY. 2003. Protein complexes and functional modules in molecular networks. Proc. Natl Acad. Sci. USA **100**: 12123–8.

121 SPRINZAK, E. AND H. MARGALIT. 2001. Correlated sequence-signatures as markers of protein–protein interaction. J. Mol. Biol. **311**: 681–92.

122 SPRINZAK, E., S. SATTATH AND H. MARGALIT. 2003. How reliable are experimental protein–protein interaction data? J. Mol. Biol. **327**: 919–23.

123 STELZL, U., U. WORM, M. LALOWSKI, et al. 2005. A human protein–protein interaction network: a resource for annotating the proteome. Cell **122**: 957–68.

124 STUMPF, M. P., C. WIUF AND R. M. MAY. 2005. Subnets of scale-free networks are not scale-free: sampling properties of networks. Proc. Natl Acad. Sci. USA **102**: 4221–4.

125 TAKAHASHI, K., N. ISHIKAWA, Y. SADAMOTO, et al. 2003. E-Cell 2: multi-platform E-Cell simulation system. Bioinformatics **19**: 1727–9.

126 TEICHMANN, S. A. 2002. The constraints protein–protein interactions place on sequence divergence. J. Mol. Biol. **324**: 399–407.

127 TITZ, B., M. SCHLESNER AND P. UETZ. 2004. What do we learn from high-throughput protein interaction data and networks? Expert Rev. Proteomics **1**: 89–99.

128 TORNOW, S. AND H. W. MEWES. 2003. Functional modules by relating protein interaction networks and gene expression. Nucleic Acids Res. **31**: 6283–9.

129 TROYANSKAYA, O. G., K. DOLINSKI, A. B. OWEN, R. B. ALTMAN AND D. BOTSTEIN. 2003. A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *Saccharomyces cerevisiae*). Proc. Natl Acad. Sci. USA **100**: 8348–53.

130 TUCKER, C. L., J. F. GERA AND P. UETZ. 2001. Towards an understanding of complex protein networks. Trends Cell Biol. **11**: 102–106.

131 UETZ, P. AND R. L. FINLEY, JR. 2005. From protein networks to biological systems. FEBS Lett. **579**: 1821–7.

132 UETZ, P., L. GIOT, G. CAGNEY, et al. 2000. A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. Nature **403**: 623–627.

133 UETZ, P. AND R. E. HUGHES. 2000. Systematic and large-scale two-hybrid screens. Curr. Opin. Microbiol. **3**: 303–8.

134 VALENCIA, A. AND F. PAZOS. 2002. Computational methods for the prediction of protein interactions. Curr. Opin. Struct. Biol. **12**: 368–73.

135 VAZQUEZ, A., A. FLAMMINI, A. MARITAN AND A. VESPIGNANI. 2003. Global protein function prediction from protein–protein interaction networks. Nat. Biotechnol. **21**: 697–700.

136 VON MERING, C., L. J. JENSEN, B. SNEL, et al. 2005. STRING: known and predicted protein–protein associations, integrated and transferred across organisms. Nucleic Acids Res. **33**: D433–7.

137 VON MERING, C., R. KRAUSE, B. SNEL, M. CORNELL, S. G. OLIVER, S. FIELDS AND P. BORK. 2002. Comparative assessment of large-scale data sets of protein–protein interactions. Nature **417**: 399–403.

138 WAGNER, A. 2001. The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. Mol Biol. Evol **18**: 1283–92.

139 WEST., D. B. 2001. Introduction to graph theory, 2nd ed. Prentice Hall.

140 WOJCIK, J., I. G. BONECA AND P. LEGRAIN. 2002. Prediction, assessment and validation of protein interaction maps in bacteria. J. Mol. Biol. **323**: 763–70.

141 WOJCIK, J. AND V. SCHACHTER. 2001. Protein–protein interaction map inference using interacting domain profile pairs. Bioinformatics **17** (SUPPL. **1**): S296–305.

142 WONG, S. L., L. V. ZHANG, A. H. TONG, et al. 2004. Combining biological networks to predict genetic interactions. Proc. Natl Acad. Sci. USA **101**: 15682–7.

143 WUCHTY, S. 2004. Evolution and topology in the yeast protein interaction network. Genome Res. **14**: 1310–4.

144 WUCHTY, S., Z. N. OLTVAI AND A. L. BARABASI. 2003. Evolutionary conservation of motif constituents in the yeast protein interaction network. Nat. Genet. **35**: 176–9.

145 YOOK, S. H., Z. N. OLTVAI AND A. L. BARABASI. 2004. Functional and topological characterization of protein interaction networks. Proteomics **4**: 928–42.

146 YU, H., N. M. LUSCOMBE, H. X. LU, et al. 2004. Annotation transfer between genomes: protein–protein interologs and protein-DNA regulogs. Genome Res. **14**: 1107–18.

32
Inferring Protein Function from Genomic Context

Christian von Mering

1 Introduction

1.1 Genomic Context – Genomes, Genes and Gene Arrangements

For function prediction, the most challenging proteins are those for which neither homology searches nor published experiments reveal any useful information. These functionally uncharacterized ("unknown") proteins are usually not known from actual observations, but instead merely inferred to exist based on sequence data from genome projects. They constitute a significant fraction of sequence databases today (uncharacterized open reading frames cover a large part of any given genome, ranging from about 20% in small, well-studied prokaryotes [17, 55] to more than 60% in large viruses [47]). While, in the future, homology searches with such uncharacterized sequences will increasingly recover at least some genes from related organisms (given the tremendous growth of genomic sequence data), these related genes will often lack functional annotation themselves.

In the absence of any other data, fully sequenced genomes remain the only useful source of information to begin characterizing such proteins (and the genes that encode them). Genome sequences can tell us a lot: which organisms do contain a gene from a particular family and which do not; how do the genes evolve; what sequences appear in physical proximity on the genomes, and much more. With genome sequences expected to be a widespread commodity soon, this information can often be remarkably useful for function prediction.

For example, the simple information on the presence or absence of similar sequences in other genomes may already narrow down the putative function of a gene to some extent: is gene X consistently found in any and all genomes queried? This may well mean that the gene performs an essential function, such as in information processing or central metabolism. Is gene Y found in a number of genomes, but never in more than a single copy per genome? This could be an indication that its function is sensitive to dosage effects. Gene Z has homologs, but its sequence appears to be changing faster than is the case

Bioinformatics - From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

for most other genes? This could be indicative of a function under changing selective pressures, such as in defense or reproduction.

Likewise, the immediate neighborhood of a gene within its chromosome can also be informative. Does a particular set of genes consistently appear next to gene W, even in phylogenetically quite distant organisms? This may indicate that the genes need to be coregulated, forming a jointly transcribed operon (in microbes), or that their consecutive order along the genome is somehow important for spatio-temporal expression regulation (as is the case for the *hox* gene cluster in eukaryotes). Regardless of the mechanisms responsible, a conserved neighborhood can often be taken as an indication that a group of genes are working together towards a common function.

Collectively, such purely genome-derived information forms the "genomic context" of a gene family. In its broadest sense, genomic context may be defined as any information on a gene that is contained in genome sequences, with respect to the occurrence and arrangement of the gene and its relatives in these genomes. Genomic context information implicitly reflects the past history of gene families and, as such, it has been shaped by a multitude of evolutionary events (mutation, selection and neutral drift). To make maximum use of genomic context for function prediction, it is essential to keep in mind that genomic context is of course relevant only when at least some aspect of its appearance can be assumed to have been under selection.

In the following, we will focus on the use of genomic context information to predict functional links ("associations") between genes; a conserved neighborhood of genes is one example for a genomic context link of this type. Predicted associations between genes are particularly useful in function prediction, because they are straightforward to interpret – if an uncharacterized gene is predicted to be associated with a better characterized gene, it is predicted to share (or contribute to) that gene's function.

It should be noted that genomic context analysis is of course not limited to function prediction of uncharacterized proteins – it can also be used to predict a functional role for proteins that do have at least a vague annotation (e.g. "likely methylase" or "transporter"), but still lack a clear functional context such as a pathway or process in which they might be involved.

1.2 Genome Comparisons Reveal Protein–Protein Associations

Three basic types of genomic context information are currently used for predicting functional associations between proteins (Table 1). They are all based on the assumption that functional partnership between proteins leaves detectable traces in the genomes during evolution. Several variations of these basic principles have been devised; these will be discussed in detail in the following sections.

Table 1

Type of genomic context	Basic principle	References
Genomic neighborhood	gene neighborhoods which are conserved, i.e. withstand the shuffling of genomes over time, are indicative of coregulation (often as operons) and may encode common functions	11, 42
Gene fusion analysis	genes that occasionally fuse to encode a single polypeptide chain (instead of two) may function together even in organisms in which they are not fused	14, 36
Gene co-occurrence	presence/absence analysis: genes that are consistently present in the same sequenced genomes (and absent together in others) may require each other for function	25, 45

The use of genomic context for predicting an association between genes can be generalized as follows. It should always begin with a systematic comparison of fully sequenced genomes, focused on the gene families of interest. One should then list all observations of the desired type (e.g. gene neighborhoods involving the families). The results should then be compared with those of other gene families (and with randomized data) in order to assess the relevance of the observation. As the three basic types of genomic context information are to some extent overlapping (shared function will often leave evolutionary traces of several types), the analysis should simultaneously employ several genomic context methods, if possible.

Generally, genomic context methods will increase in predictive power as more and more genomes become available, and they are applicable to any class of gene regardless of the type of proteins they encode (enzymes, structural proteins, regulators, etc.). Various implementations exist for each type of genomic context, and these differ significantly in technical and conceptual detail, and in computational cost.

1.3 Prerequisites for Genomic Context Analysis

Most proteins, but not all, are amenable to genomic context analysis. For a successful prediction, a number of prerequisites must be met: (i) the protein in question should be from an organism whose genome sequence is known, (ii) other sequenced genomes must be available (the more the better) at various phylogenetic distances relative to the query genome, (iii) all those genome sequences should ideally be finished, i.e. consist of high-quality contigs with little mis-assembly or gaps, and (iv) the gene in question must have counterparts in at least some of the other genomes (genomic context is not applicable to "molecular orphans", i.e. genes without any detectable relatives [18]).

Defining counterparts of a gene in other genomes is a key step for genomic context analysis. This is because it only makes sense to compare genomes if we can assign which part in one genome corresponds to which part in the other. What exactly is meant by a "counterpart" of a gene in another genome? Since genomic context analysis is based on detecting the long-term effects of shared function on genome evolution, the gene and its counterpart should be defined as having the same function. However, how can we know whether two genes have the same function – since function is what we aim to predict in the first place?

Luckily, we do not have to know the actual function of either gene to check whether they may be functionally equivalent; it is enough to test whether they are homologous or, better, orthologous. Generally, if two genes are homologous (i.e. of common descent), then they have a certain likelihood of performing the same function, especially if their sequence similarity is high (for low similarities, this is of course less certain – the actual percent identity cutoffs that should be applied are under debate [50, 56]). One particular subtype of homology, called *orthology* [15, 16], leads to the most reliable assignments of functional equivalence and it is often used in annotating newly sequenced genomes. Two sequences in two different organisms are defined to be orthologous if they trace back to a single sequence in the last common ancestor organism, as opposed to being paralogous for the case when they were already separate entities in the last common ancestor (see also Chapter 4). Because orthologs trace back to a single sequence at the time of speciation, they have a higher chance of having retained an identical function, even if their actual sequence similarity is low.

Orthology of two genes is always merely a hypothesis, and it can be difficult to decide which, if any, among the many homologs is the true ortholog. Fortunately, several databases exists that contain globally precomputed orthology predictions for a range of complete sequenced genomes [40, 55]; where such data is not available, homology can still be used instead (see Chapter 30).

1.4 How Specific are the Inferred Functions?

Genomic context analysis works by searching for putative effects of selection. A conserved gene arrangement, for example, may have been formed and retained because it provides a selective advantage for the organism to keep functional partners in close vicinity. Likewise, if one gene is lost from a genome because it is apparently no longer selected for, then many of its immediate functional partners may also be no longer under selection and they might be lost as well (the latter assumption forms the rationale behind the "gene co-occurrence" approach).

At the molecular level, the vast majority of large-scale genomic features are shuffled randomly, but those that are under selection can be made to stand out when enough genomes are being compared. As selection, ultimately, works at the level of the entire organism (or population), the specificity of the function prediction for any particular gene is inherently limited. A genomic context association prediction should always be thought of as "genes X and Y apparently share a common selection pressure". This is then translated into "genes X and Y participate in the same functional process". This translation is a leap of faith – it assumes that selection can indeed act on specific, separable functional processes and that these will somehow coincide with our definition of pathways or processes (or protein complexes, regulons, etc.). In some cases, the situation is pretty obvious, e.g. the biosynthesis of a complex organic molecule may involve several specific enzymes whose only purpose is catalyzing one step each of the reaction. The final pathway product is what is actually needed by the organism and is presumably what is being selected for – so that all enzymes needed to produce it share indeed a similar spectrum of selection during genome history. They form a unit and this may be detectable by inspecting genomes.

Of course, genomic context analysis will not normally be able to uncover the molecular detail of the actual enzymatic reactions. In other instances, it may not even be possible to identify clear functional units that could be selected for – some proteins may function in more than one pathway or may function autonomously without any direct partners. Nevertheless, the evolutionary approach taken in genomic context analysis has the great advantage of being independent of human classifications and biases – whatever evolution deems important enough to be selected for as a unit, will potentially be detectable as a group of genes, and can be interpreted as functional partners.

2 Gene Neighborhood

2.1 Conserved Neighborhood versus Simple Synteny

The physical proximity of genes in the genome (*gene neighborhood*) is the simplest form of genomic context information – it is routinely used in microbiology, even since long before the advent of the first complete genome sequences. In prokaryotes, gene neighbors often form operons [26]; operons represent the first recognized example of genes neighboring each other for functional reasons. In operons, several neighboring genes located on the same strand of the DNA form a polycistronic transcription unit (i.e. they are transcribed together as a single mRNA), presumably because they can be regulated more efficiently in this way. Usually, the genes in an operon will work together towards a common function – in microbiology, this has often been exploited for placing uncharacterized genes into their functional context (e.g. Ref. [22]).

However, not all genes that are found in physical proximity will be part of a polycistronic operon. Some genes may be neighbors simply because of a recent genome rearrangement which happened to place them next to each other. To some extent, the intergenic distance between genes can be used to distinguish true operon members from accidental neighbors, but this alone is not very reliable (having a roughly 20% error rate [38]). Fortunately, comparative genomics can help: true operons should have a higher chance of being observed in other genomes as well, whereas "chance" neighbors should be separated as fast as they are formed. Thus, any gene neighborhood that is observed in several genomes (a "conserved neighborhood"), leads to the prediction that its genes are forming a true coregulation unit and are thus likely to function together.

Unfortunately, the above is true only when the genomes in question are sufficiently distant phylogenetically – otherwise even chance neighbors can be observed repeatedly because of genome synteny. *Genome synteny* is defined as a broad similarity between homologous genome segments in two species, and it usually occurs simply because insufficient time has passed to shuffle the genomes and their gene arrangements through random events such as inversions or translocations. The timescale of genome synteny "decay" is an important factor for neighborhood analysis (because within that timescale, repeated observations of a neighborhood cannot be considered independent). For prokaryotes, synteny is usually restricted to species of at least the same phylogenetic order, e.g. within the Vibrionales, or the Enterobacteriales, but not between these two (Figure 1). For all types of gene neighborhood analysis, it is essential to determine and exclude the effects of genome synteny. Since synteny is usually assumed to evolve (and decay) neutrally with time, it can best be measured by tracking the fate of genes that lack any reasonable functional link – if they are neighbors, they are expected to be shuffled away at random. Such neutral neighbors could be, for example, well-characterized genes working in totally unrelated cellular functions, or gene neighbors arranged in a tail-to-tail orientation (Figure 1). Tail-to-tail arrangements are thought to be least likely coregulated, or functioning together, since they are transcribed from opposite strands and from distant promoters.

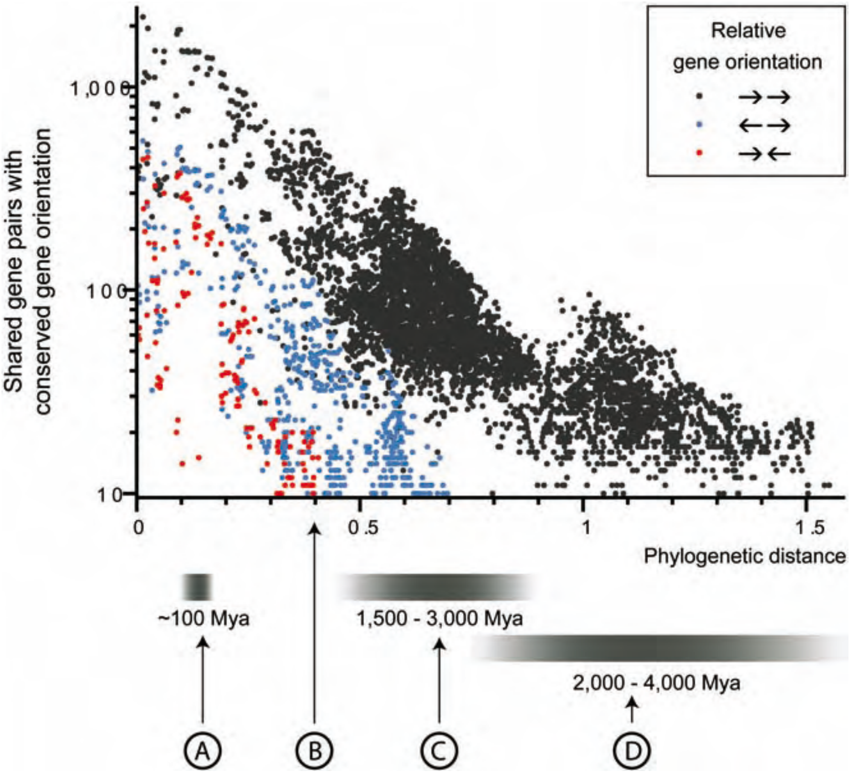


Figure 1 Decay of genome synteny with increased phylogenetic distance. Each dot in the graph represents a comparison between two fully sequenced prokaryotic genomes. The x-axis describes the phylogenetic distance between the two genomes [32] and the y-axis indicates how many gene neighborhoods are still detectable in both organisms (plotted separately for each of the three possible gene-to-gene orientations). The letters on the x-axis indicate the approximate time points of some representative evolutionary events (A: split between *Escherichia* and *Salmonella*; B: time point at which synteny is largely lost; C: split of main bacterial groups, such as Gram-positive or Gram-negative; D: Split of domains Bacteria and Archaea). The tail-to-tail orientation decays quickly with time (red dots); whereas in particular the codirectional neighbors (black dots) can be conserved over very large phylogenetic distances. This difference indicates that the three possible gene-to-gene arrangements are under different levels of purifying selection. If one assumes that tail-to-tail gene arrangements are not selected for at all, then these provide a background measure for the level of neighborhood that is expected simply due to genome synteny. (Modified from Ref. [31], with permission.)

2.2 Operons and "Über-Operons"

Operons are the most stable gene arrangements in prokaryotic genomes; however, even operons do occasionally break up, re-form or change the relative order of their constituent genes. Any given conserved gene neighborhood, therefore, even when found in a wide array of species and clearly selected for, can be "broken" in a particular species – the genes may be present in the genome, but in scattered locations, no longer forming a gene neighborhood. This does not necessarily mean that the genes are no longer sharing a function; it may simply be the result of an episode of extensive genome shuffling or of relaxed constraints on the regulation of the pathway. Over time, and with more and more genomes available, this leads to a situation in which only very few operons remain totally unchanged in all organisms studied. However, many of the changes in gene neighborhoods are conservative [34], i.e. operons are often re-formed (although possibly with differences in gene orientation or composition). For any given functional system, various partial or complete gene neighborhoods can be observed in fully sequenced genomes today; these can be thought of as variants of a prototypic, often larger, "über-operon" [34] (Figure 2). If each subpart of an über-operon is well-supported by repeated observations, then the entire über-operon can be assumed to encode functionally associated proteins.

This generalization is very valuable for function prediction – it means that, for any given species, a large number of functional links can be predicted, even for gene pairs that are not actually neighbors in the query species itself. For example, the arginine biosynthesis genes *argB* and *argC* are not neighbors in the genomes of cyanobacteria, yet they can be linked functionally, based on their recurring neighborhood – in various orientations – in many other bacterial clades.

The über-operon concept also implies that neighborhood analysis is not restricted to immediate neighbors: the two gene families *argJ* and *argF*, for example, are from a conserved neighborhood in most genomes, even though they are never direct neighbors (i.e. there is always at least one intervening gene between them in the genomes we know so far). In fact, if one assumes that neighborhood relations are fully transitive, this could even lead to association predictions for gene pairs that are never close to each other in any of the genomes: if gene A consistently forms a neighborhood with gene B and gene B with gene C, but genes A and C are never neighbors, then they could still be predicted as functional partners. Of course, as is the case with neighborhood analysis in general, any such observation needs to be carefully checked against two null-hypotheses: (i) that the observed scenario may have arisen by pure chance and (ii) that the observed scenario is merely due to genome synteny (see Section 2.1). Fortunately, both null-hypotheses can be tested: (i) gene arrangements can be randomized *in silico* to verify that it is unlikely to observe a given neighborhood by chance and (ii) the genomes to be analyzed can be made nonredundant, such that closely related genomes displaying remnants of synteny are considered as a single item only.

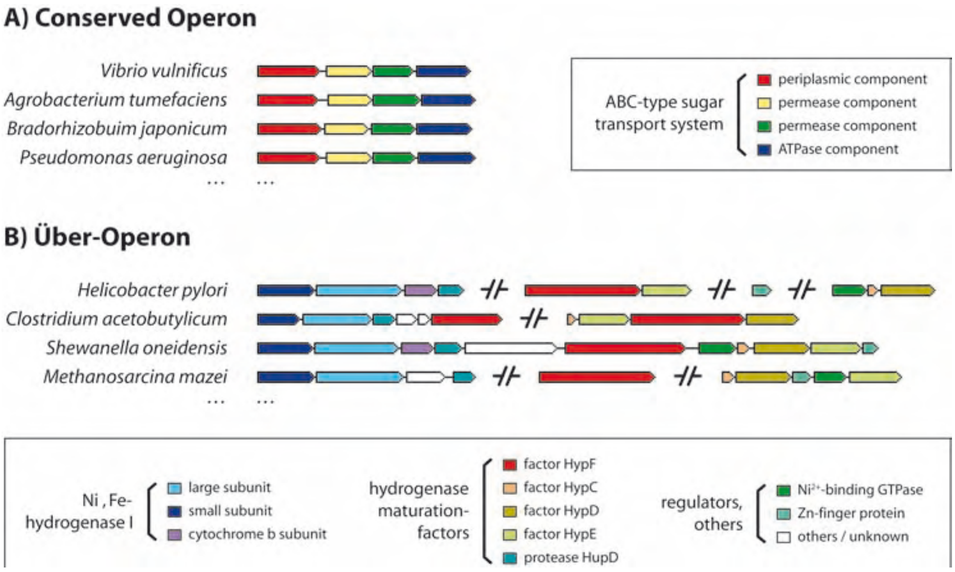


Figure 2 Typical genomic arrangements of functionally associated genes. Each colored section symbolizes one gene (i.e. an open reading frame) in a prokaryotic genome. Genes are shown as neighbors wherever they are encoded on the same DNA strand and are in close proximity (separated by fewer than 300 bp). Genes that are not neighbors are shown separated by break symbols, denoting a large area of intervening DNA that may contain many unrelated genes. The top panel shows the simplest case of informative neighborhood – a conserved neighborhood, found in identical composition and arrangement in many diverse bacterial genomes. These genes are likely to form a cotranscribed operon in many of the organisms. In contrast, the lower panel shows a more complex gene neighborhood, where the genes are neighbors in various different arrangements, forming one or several operons. This is more frequently observed than the ideal situation shown in the top panel; it is nonetheless indicative of a larger system of genes sharing a function (in this case, a Ni,Fe-hydrogenase is shown with its associated regulators and maturation factors).

2.3 Divergently Transcribed Gene Pairs

When scanning genome databases for repeatedly occurring gene neighborhoods, the most frequently observed arrangement is the collinear arrangement, in which the conserved neighbors are located on the same DNA strand and are transcribed in the same direction. This is not surprising – at least in prokaryotes, most genomes are enriched in stretches of consecutive genes transcribed in the same direction (partly because of operons, but also in part because of simple strand biases [49]). Thus, even a randomly selected gene pair has a higher chance of being observed repeatedly, if its arrangement is collinear. Apart from the collinear arrangement, two other arrangements of gene neighbors are possible: divergent (i.e. neighbors oriented in

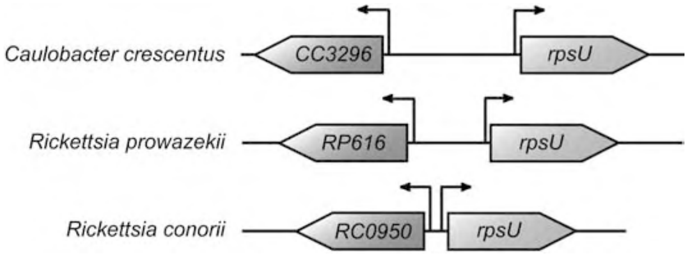


Figure 3 Example of a conserved, *divergently* arranged gene neighborhood. The two gene families (red, green) are found next to each other, in a divergent orientation, in three microbial genomes. The family shown in red contains previously uncharacterized genes weakly homologous to members of the TetR family of transcriptional regulators. The family shown in green contains genes encoding ribosomal proteins of the S2 type (rpsU). This observation leads to the suggestion that genes of the red gene family serve as regulators of ribosome production and of protein S2, in particular. The red gene family has orthologs in higher eukaryotes (including humans). These are not in any divergent gene neighborhood, but they may still perform the same function in humans (possibly in mitochondria, which are derived from prokaryotic ancestors). Thus, we have here an example of a genomic context prediction relevant to human biology, although the initial observation was made in prokaryotes. (Simplified from Ref. [31], with permission.)

a head-to-head fashion, transcribed away from each other) and convergent (i.e. neighbors orientated in a tail-to-tail fashion, transcribed towards each other). Interestingly, among repeatedly observed neighborhoods, the divergent arrangement is much more frequently seen than the convergent arrangement (Figure 1), although both are *per se* about equally frequent in any given genome. This suggests that, to some extent, even divergent gene neighbors are preferentially retained during genome evolution – presumably because some of them are functional partners maintained by selection. Indeed, this observation forms the basis for a recent extension [31] of the conserved neighborhood method: not only collinear neighbors, but also divergent neighbors are predicted to encode functional partners, if their arrangement is observed sufficiently often (see Figure 3 for an example).

It is not yet fully understood why divergent arrangements should provide a selective advantage, but one possibility is that they facilitate coordinated regulation, either through shared promoters or through mechanistic "cross-talk" between the neighboring loci [48]. Indeed, analysis of expression data for neighboring gene pairs suggests that divergent pairs have a higher chance of being coexpressed than convergent gene pairs [31]. Coregulation through a shared promoter might have the advantage of being very "immediate" (no intermediate steps are required for the regulation to take effect); tightly coupled coregulation might be particularly advantageous if one of the partners were a regulator itself, controlling the other partner (e.g. in a feedback loop). Korbel and coworkers have observed that the latter might indeed occur frequently: in many of the pairs, one gene encodes a transcription factor, regulating the other gene [31]. Divergent gene pairs may be a widespread phenomenon – they have been observed not only in prokaryotes, but also in higher eukaryotes such as humans [1].

2.4 Gene Neighborhood in Eukaryotes

Analyzing gene neighborhood has long been routine in prokaryotes – both in bacteria and in archaea. Prokaryotes have very compact genomes, thus their genes are naturally in very close proximity, and the known prevalence of operons shows that tight coregulation of neighboring genes is commonplace in prokaryotes. But what biological meaning does gene neighborhood have in eukaryotes?

In particular, multicellular eukaryotes can have a very low gene density, with noncoding sequences vastly outnumbering coding sequences (genes) in the genome. Eukaryotic genes are usually transcribed separately, each forming a single transcription unit, and not as longer, polycistronic transcripts covering several genes at once (operons). However, many unicellular eukaryotes do have quite high gene densities, and some basal eukaryotes, as well as nematodes, even have polycistronic transcripts [5,58]. Does this mean that neighborhood analysis can easily be extended to eukaryotes?

There are certainly some well-characterized examples of genes whose orientation and neighborhood on the genome is important, and presumably maintained by selection. A famous example is the "*hox* cluster", a set of homologous transcription factors usually found in a strict spatial ordering on the genome of animals. Each gene serves as a molecular switch for providing "identity" to a particular body part during development and the body parts in question are often arranged in the same spatial order as the genes on the genome. This gene arrangement bears all the hallmarks of a conserved gene neighborhood, potentially useful for genomic context analysis: the genes are present in a wide range of organisms (from insects to man), their neighborhood is consistently observed (with only a few exceptions) and the neighborhood is far more conserved than most other neighborhoods at equivalent phylogenetic distances. Indeed, had the *hox* cluster not already been appreciated through genetics experiments, genomic context analysis and the current availability of genomes would have easily revealed it as a special gene arrangement.

Unfortunately, only very few other examples seem to exist (in eukaryotes) of genes apparently maintained in each others' neighborhood. These include genes encoding histone subunits, as well as certain immunoglobulin gene clusters. In general, however, gene neighborhood in eukaryotes appears free to evolve, with few apparent evolutionary constraints. This is bad news for genomic context analysis: only when evolutionary constraints can be presumed to exist, does genomic context hold information about gene function. At least for higher eukaryotes, there seem to be too few constraints for neighborhood analysis to be useful at large – for basal eukaryotes, however, the jury is still out (because not many genomes have been analyzed so far).

That notwithstanding, neighborhood analysis can still be informative for eukaryotes, albeit indirectly – for all those eukaryotic genes for which clear orthologs can be identified in prokaryotes, any neighborhood that is observed in prokaryotes allows a prediction for the eukaryotic genes as well: functional partnerships are presumably maintained for much longer time than a particular gene neighborhood. A particular metabolic pathway, for example, might be recognizable through genomic context analysis in prokaryotes – then the genes will most likely form a pathway even in eukaryotes, although their mode of regulation may have changed. This can lead to the discovery of the function of hitherto uncharacterized genes, in higher eukaryotes and even in humans (e.g. Ref. [6]).

3 Gene Fusion

3.1 Gene Fusions and Gene Fissions

Among the many types of mutations that can affect genes and genomes, some are capable of changing where a gene starts or ends. A point mutation that introduces a stop codon, for example, might accidentally shorten a protein, whereas the inverse event (a stop codon changing to a valid codon) may lengthen it. Often, length changes will effectively destroy the gene, but in some instances the gene will remain functional. In extreme cases, a gene can even break up into two or more pieces that each remain functional (gene fission) or it may occur that two genes that have so far been separate become one entity, encoding a single polypeptide chain (gene fusion).

With respect to present-day genome sequences, successful gene fusions or fissions in the past led to very similar observations: two or more gene families that appear as separate genes in some of the genomes, but as longer, single genes in other genomes. Whether the underlying event was a gene fusion or gene fission is often difficult to discern and can sometimes only be inferred through careful phylogenetic analysis. Such analyses have been executed systematically [33, 51], and it has been observed that both events indeed occur during evolution and that fusions "survive" noticeably more often than fissions.

For genomic context analysis, both types of events hold information about protein function: in the case of the fission, the two separate gene families were once a single family (and thus were presumably part of the same function). In the case of the fusion, the two gene families apparently can occur together, i.e. they can tolerate being "tied" to each other; this means that the encoded proteins do not hinder each other sterically, are capable of working in the same cellular localization, and can function under the same transcriptional and translational regulation. This tolerance of course merely means that the fusion architecture is not selected against. However, a fused gene that lasts over long evolutionary time is probably not only neutral in the above sense, but may even be beneficial for the cell – otherwise it would have a high likelihood of breaking apart again into its constituents. Thus, whatever the underlying scenario, two separate gene families that are observed as a composite gene in some genomes can be predicted with some certainty to function in the same process or pathway. This basic observation was soon realized after the advent of the first complete genomes and is the basis for one of the three major genomic context techniques – the *gene fusion* method [14, 36].

For the gene fusion method, genomes are systematically scanned for open reading frames that seem to encompass more than one gene family. The gene families in question should be clearly defined as separate entities in other genomes, i.e. they should appear as distinct genes and they should not be homologous. If a candidate for a fused gene is found, it should be verified that the observation is not simply due to a sequencing error or a mistake in genome assembly. Next, it should be checked whether the fusion can be confirmed independently, i.e. rediscovered in another genome. This strengthens the case, because a single observation could be due to a chance event, such as a recent translocation or inversion (even when correctly sequenced). In addition, one should search for other fusion scenarios involving one of the two families: in real-world cases, a gene that undergoes fusions often will do so with more than one partner – usually all from the same pathway (e.g. Figure 4). Once a gene fusion (or fission) is observed and confirmed, a prediction ensues that the gene families in question have a functional relationship, even in all those genomes where they occur separately (not fused). This prediction is particularly strong when it can be demonstrated that the fused parts are orthologous to the distinct parts in the other genomes (as opposed to being merely homologous). This usually is the case when the fused parts are the only instances of the respective gene families in the genome containing the fusion gene.

3.2 Functional Implications

Gene fusions are distinct from all other types of gene context information, in that they entail a covalent link between two (or more) normally separate proteins. This implies a close, physical partnership between the proteins

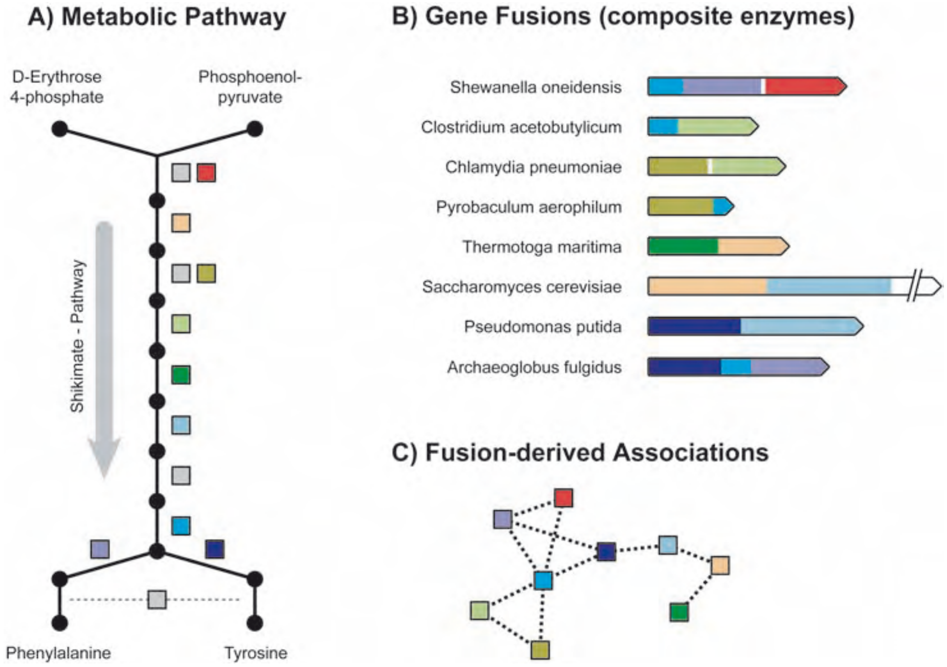


Figure 4 Gene fusion scenarios within a metabolic pathway. (A) The shikimate pathway is a biosynthetic metabolic pathway employed by plants and bacteria to generate aromatic amino acids (from precursors taken from carbohydrate metabolism). The colored symbols denote enzymes needed to perform the reactions in the pathway. (B) A selection of gene fusion scenarios observed for this pathway in completely sequenced genomes. Note that fusions occur in various combinations and do not necessarily involve direct pathway neighbors. (C) Association network derived from the fusion architectures. Note that the network connects all but four of the enzymes subunits of the pathway.

even when they are not part of a fusion; they possibly form part of a larger protein complex. In the cell, many important functions are performed by large complexes consisting of several tightly bound protein subunits (e.g. the ATP synthase complex, the fatty acid synthase complex or the pyruvate dehydrogenase complex). In many cases, the individual subunits of such complexes cannot function at all on their own – only the fully assembled complex is functional. In other cases, the individual subunits can to some extent fulfill their role in isolation, but do so more efficiently in a complex (where they may pass on substrates, akin to an assembly chain [61], or be more efficiently transported or integrated into a particular cellular location). For genes encoding the proteins in such macromolecular assemblies, it may be beneficial to fuse, enabling a tighter coupling and better coregulation. On the other hand, the synthesis of proteins may become costly and error prone when they become too large, especially in organisms under extreme conditions

[51], which may effectively limit the process. These balancing forces lead to a situation where many complexes or functional systems contain a certain number of fusion genes, but not necessarily the same fusion architectures in each organism (Figure 4).

Surprisingly, some protein complexes exhibit very few gene fusions (e.g. the ribosome) and it is unclear why this is the case. It could of course be due to special requirements in protein folding or complex assembly, or it may indicate that the complex is highly optimized and cannot tolerate many changes (consistent with this is the fact that the ribosome shows a very high level of sequence conservation). In general, gene fusion events are most often observed for metabolic enzymes, e.g. in *Escherichia coli* three-quarters of the total gene fusions affect metabolic genes [57].

3.3 Gene Fusions versus Domain Analysis

Some gene families are extremely prolific with respect to gene fusions. They participate in a large variety of fusion events, often with a diverse array of partners. Many of these promiscuous families are involved in signal transduction (e.g. histidine kinases or methyltransferases). Not only do they frequently participate in gene fusions, but they are also widespread in general, being present in several copies in most genomes. For the gene fusion method, these gene families pose a challenge. Due to their fast evolution and high copy number, it can be difficult to trace their phylogeny and to decide which of the genes within such a family are orthologous, for a given set of genomes. Without clear orthology information, however, a single fusion event involving such a gene family would lead to a prediction concerning all members of the family. For example, a fusion event involving a histidine kinase (such as the fusion of a certain histidine kinase to a likely Na^+/proline symporter in *Rhodopseudomonas palustris*) would lead to the prediction that all histidine kinases function in the respective process (proline import), which is clearly not the case.

In fact, these gene families are seldom seen in isolation, which is why they are sometimes not referred to as genes in their own right, but instead as fragments of larger genes; they often encode so-called *domains*. Domains are autonomously folding building blocks of proteins, which are frequently rearranged to form multi-domain proteins of significant architectural flexibility. They are frequently observed in areas such as signal transduction or immunity, in which novel proteins are quickly formed and changed during evolution.

Conceptually, the rearrangement of domains in order to form new proteins is of course simply a series of gene fusion and fission events – this is why the distinction between “domain analysis” and “gene fusion analysis” is not always clear. In general, the difference between both approaches lies in the precision of the phylogenetic information sought after: for gene fusions, we would like to know precisely, which genes (in which organisms) led to the fusion event; the goal is to be able to describe the phylogenetic relation of the various players. In contrast, for domain analysis this is usually not of interest, one does not ask what role the domains played before the fusion or where they came from. This is because domains are assumed to carry a certain basic molecular function, but are thought to take on a specific biological role only in conjunction with other domains, in a concrete protein.

4 Gene Co-occurrence

4.1 Phylogenetic Profiles

The gene complement in any given genome is subject to change over evolutionary timescales. Some genes may turn into pseudogenes and eventually be lost completely; others may newly enter the genome (for example through horizontal gene transfer from other species). Genes may also be invented de novo from previously non-coding genetic material; others may mutate beyond recognition and take on a new functional role, effectively becoming "new" genes. However, all of these changes are obviously constrained, and limited, by the effects of selection – leading to the general picture where changes in genome content are known to occur, but are not frequent enough to completely scramble genome identity [52].

As the gene complement is constrained by selection, changes in gene content are to some extent informative about changes in biological function. Any gene that is permanently lost, for example, indicates that a function may no longer be needed by an organism (such as the loss, in mammals, of genes to make a hard egg-shell; presumably this is a consequence of the internalization of their embryos [24]). The same is true for genes that are newly gained: many bacterial pathogens have recently acquired "new" antibiotics resistance genes, in a clear response to selection pressures mounted by humans. In the long run, the presence or absence of a gene in a genome thus appears to depend on, and reflect, its function.

As many genes work in teams, one would expect the same to be true for entire groups of genes – their presence should depend on their shared function and they should tend to be either present together or absent together in any given genome. Indeed, the presence of any particular gene increases the likelihood of finding its functional partners in the same genome: the presence/absence pattern of a gene among fully sequenced genomes (its *phylogenetic profile*) is often similar to the profile of its functional partners. This has been exploited as the basis of one of the major genomic context prediction methods: the *gene co-occurrence* approach. In this approach, phylogenetic profiles of all genes are compared, and if two or more genes show similar profiles, then they are said to "co-occur" and are predicted to be functional partners [25, 45]. This is a very powerful approach, because it does not depend on the actual genomic arrangement of genes or on their mode of transcriptional regulation – the simple presence of the gene anywhere in the genome provides the signal.

In real-world applications of this principle, it is important to take a close look at the phylogenetic profile of a gene family of interest before conducting a search, because not all profiles are equally "informative". For example, a gene family that is known to be present in all genomes has a profile of "full presence". Such a profile is not very specific; many other gene families have this profile. It presumably merely signifies that the gene family has an essential function needed by all organisms; however, because there are several essential functions (replication, transcription, translation, energy metabolism), it does not mean that all proteins with a "full presence" profile have the same function. Another pattern which is not very informative is the case that a gene occurs only in a single genome, i.e. it is a "molecular orphan". Usually, every genome has a number of such rare or fast-evolving genes, but this does not mean they are all functionally related. Another class of patterns that carry only little information encompasses those patterns that fully cover a single clade (or phylum) of organisms. For example, a gene that is present in all archaea, but nowhere else, has a phylogenetic profile that is neither "full presence" nor "molecular orphan". Yet, it is still not very informative, because many genes are known to have such a pattern (there is a significant length of evolution preceding the last common ancestor of extant archaea, during which a number of genes were invented that are now common to all archaea).

What, then, is an informative pattern for phylogenetic profile analysis? Basically, any pattern that appears "patchy". Ideally, the gene family should cover wide range of organisms, but it should not cover them completely, such that there are many instances in which the genes were presumably lost and/or transferred. In cases where such a "patchy" presence/absence pattern is matched by a second gene family, then a confident prediction for a functional link between the two families can be made, because the observation is unlikely to have arisen by chance (e.g. Figure 5). With more and more genome sequences known, the number of gene families that have a somewhat "patchy" profile is rising – fewer and fewer families show a "full presence" or otherwise entirely uninformative pattern.

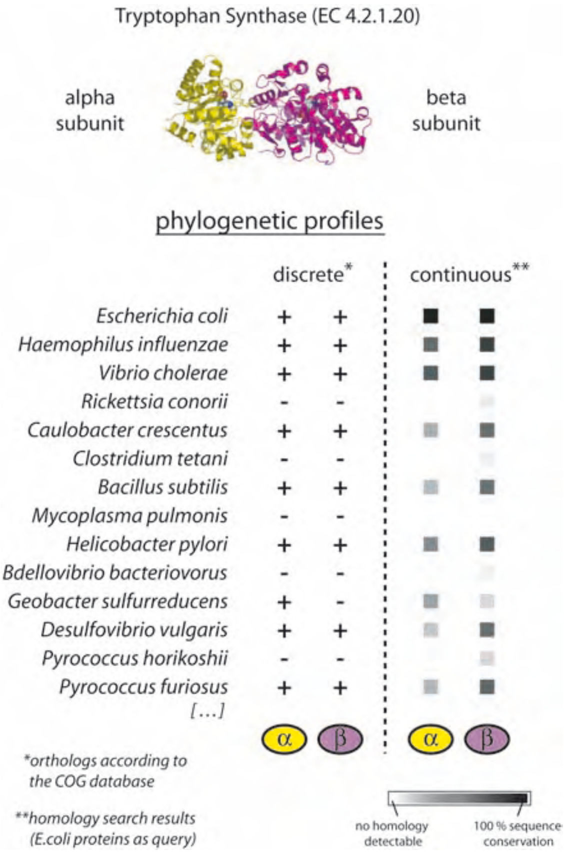

Tryptophan Synthase (EC 4.2.1.20)
alpha subunit
beta subunit
phylogenetic profiles
discrete*
continuous**
Escherichia coli + +
Haemophilus influenzae + +
Vibrio cholerae + +
Rickettsia conorii - -
Caulobacter crescentus + +
Clostridium tetani - -
Bacillus subtilis + +
Mycoplasma pulmonis - -
Helicobacter pylori + +
Bdellovibrio bacteriovorus - -
Geobacter sulfurreducens + -
Desulfovibrio vulgaris + +
Pyrococcus horikoshii - -
Pyrococcus furiosus + +
[...]
α β
α β
*orthologs according to the COG database
**homology search results (E.coli proteins as query)
no homology detectable
100 % sequence conservation

Figure 5 Two proteins known to function together and the "phylogenetic profiles" of their genes. (Top) The two subunits of the tryptophan synthase enzyme are known to tightly bind each other and to jointly perform their function in tryptophan biosynthesis. (Bottom) The genes encoding these two proteins are usually either both present in a genome or both absent (i.e. their phylogenetic profiles are similar). Profiles can be either shown as discrete presence/absence profiles (left) or as continuous profiles (right, showing in this case the sequence similarity of the closest homolog in BLAST searches). Notice that the profiles of the two proteins are quite similar, even though the profiles are nontrivial ("patchy") and even though the proteins are not evolutionarily related.

4.2 Discrete versus Continuous Profiles

For the gene co-occurrence technique, it is particularly critical to have a reliable assignment of whether or not a gene family is present in a particular genome. Mistakes in this assignment will affect this method more than the gene-neighborhood or gene fusion methods, because the latter rely on gene-to-gene arrangements that are detectable even when the occasional gene copy is overlooked or erroneously included. In contrast, the co-occurrence technique relies on the gene presence itself as the signal and any error in assigning gene presence will introduce noise into the phylogenetic profiles.

Even when restricting the analysis to genomes that have been carefully finished and are of high quality, it can still be difficult to judge whether a functional counterpart of a gene is present or absent in a genome. As discussed in Section 1.3, the most reliable indicator for functional equivalence is "orthology", i.e. the situation that the genes in question presumably all trace back to a single ancestral sequence in the last common ancestor organism. However, orthology is always merely a hypothesis and can be difficult to assign in an automated fashion; possible errors are accidental oversight of true orthologs or unnecessary inclusion of paralogs (i.e. genes related through a gene duplication event, instead of a speciation).

An alternative measure of functional equivalence is simply the degree of sequence similarity between two genes – the higher the sequence similarity, the more likely the two genes have retained their function and are equivalent. While this measure clearly is not a good substitute for orthology, it is much easier to automate and just requires an arbitrary similarity cutoff below which a gene family is assigned as "absent".

Due to the general difficulties in reliably assigning gene presence, two types of phylogenetic profiles have been introduced: "discrete" profiles and "continuous" profiles. Discrete profiles are those that simply mark the presence or absence of a gene family in each genome; they essentially take the form of a vector with two types of elements: 0 (for absence) and 1 (for presence). Continuous profiles, in contrast, do not attempt to decide whether or not a gene is presence, but simply provide a measure that is thought to correlate with functional equivalence (e.g. sequence similarity). Continuous profiles take the form of a vector with values ranging from 0 to 1; a value of 0.8 might for example signify a sequence identity of 80% or any other quantitative measure pointing towards a high chance of functional equivalence.

4.3 Profile Distance Measures

With more and more genomes being sequenced, very few gene families are found to be present in exactly the same set of organisms (i.e. to have the same, discrete phylogenetic profile). This is true even for gene families known to be tightly associated functionally. In part, this lack of agreement can be attributed to technical limitations such as occasional gaps in genome sequences (or errors in orthology assignment), but it is also due to the stochastic nature of genome evolution itself: phylogenetic profiles are shaped by the inheritance, loss and/or horizontal transfer of genes in and between genomes. For genes sharing the same function, the overall outcome of this will be roughly similar (due to selection), but the individual evolutionary events are still stochastic.

A group of genes forming a metabolic pathway, for example, may become dispensable and will be eventually lost from a genome, but the loss takes time and may proceed in steps, and many of the genes may initially still be detectable as pseudogenes. Any phylogenetic profile observed at a given point in time thus contains a certain measure of biological and technical "noise".

This necessitates the use of distance measures in comparing profiles, in order to assess which profiles are most similar to a given query profile. A variety of different distance measures can and have been used, and since there is no formal and fully parameterized model of genome evolution, the choice of distance measure is determined empirically. The choice is guided mainly by the performance of the various distance measures in recovering known functional associations from the comparisons, and the choice may depend on the set of known reference associations, the number of genomes analyzed and the type of question asked.

For discrete profiles, a suitable distance measure is mutual information, a measure from information theory that describes the amount of information one profile contains about the other profile; two identical profiles have a large mutual information, whereas random, independent profiles will have a mutual information of zero. Another useful measure for discrete profiles is the hamming distance, i.e. the simple count of the positions in which the two profiles disagree. In the case of continuous profiles, useful distance measures are the Euclidian distance or the Pearson correlation coefficient and vectors may be normalized before the distance is assessed.

In any case, the distance measure will provide a ranking, for any given query profile, of other profiles that best resemble it. Among those, known and novel functional partners should be enriched. Distance measures in general also enable the design of a scoring system, because they provide a quantitative measure which can be benchmarked with regard to its predictive power, using previous knowledge. This predictive power can then be expressed as a *confidence estimate* – a certain distance is assigned a certain likelihood of corresponding to a useful prediction.

4.4 Tree-based Methods

Conceptually, any simple distance measure between two phylogenetic profiles is ignorant of the evolutionary relationships between the species. However, it is the evolution of species (and the ensuing changes in their genome content), that forms the conceptual basis for the gene co-occurrence method. Essentially, the "unusual" evolutionary events within the history of a gene family are what make its profile unique: gene loss, gene duplication and (horizontal) gene transfer.

Ideally, therefore, it should not be the phylogenetic profiles themselves that are compared, but the inferred evolutionary events that happened in the past of the two gene families: did these events have some influence on each other (preferentially happening in concert) or did they occur independently? Focusing on past evolutionary events would automatically provide a way to recognize "uninformative" profiles: uninformative profiles would be those that can be explained with very few evolutionary events. A "full presence" profile, for example, requires only a single event (a single gene birth before the last common ancestor). Apart from that event, the profile can be explained by standard vertical inheritance. A single event, however, is certainly not enough to propose a shared function for any other family sharing that event.

Unfortunately, it is difficult to infer with some certainty what happened during the evolution of a gene family, and when it happened. Usually, this involves the alignment of the sequences, and the generation of a phylogenetic sequence tree (see Chapter 4). This tree is then compared to the previously known organismal tree, and any deviation is taken as an indicator for an unusual evolutionary event. Parsimony analysis is then used to determine the minimum number of events necessary to explain the observed patterns. This whole process is difficult to automate, however, and unsupervised procedures are unlikely to result in accurate alignments and reliable evolutionary inferences. Nevertheless, including tree-based information represents a very promising avenue for analyzing gene co-occurrence and early attempts to do this are promising: trees can be used to reduce the bias and redundancy in completely sequenced genomes, before comparing profiles [59], or the trees can be compared directly to infer the interaction specificity for interactions involving large multigene families (such as receptors and their ligands [46]).

4.5 Anti-correlated Profiles

Remarkably, gene co-occurrence is not the only way in which functionally associated gene families can manifest themselves in terms of genome content. In some instances, functionally associated genes do not co-occur at all, but actually appear to avoid each other. This can happen when the two families in question are not merely functionally associated, but indeed have the very same, identical function. In this case, the reason for the apparent avoidance is simple: having the same function, the genes can effectively replace each other and a situation where both of them would be together in the same genome is of no evolutionary advantage, so usually only one of them is kept. Which family ends up in which genome is probably largely random, but the overall outcome is that the two families have phylogenetic profiles that are essentially complementary to each other.

In order to appear as two distinct families in the first place, the involved proteins should have distinct sequences, and display little or no sequence similarity. If the families are not similar in sequence, how can they still have the very same function? This is usually attributed to a process called *convergent evolution*, in which a certain molecular function has most likely evolved twice, independently, from unrelated sequences (or at least from sequences that have diverged significantly before converging again functionally).

Well-known examples of anti-correlated profiles have been described, such as the case of the complementary occurrence of class I and class II lysyl-tRNA synthase genes [19]. In principle, anti-correlations can provide a way to extend the usefulness of phylogenetic profiles: for any given query gene, one should not only search for the best-matching profile, but also for the profile that best complements it. In this way, gaps in metabolic pathways can be closed and new functions assigned to previously uncharacterized sequences acting as analogous enzymes (e.g. Ref. [39]). Overall, of course, convergent evolution and functional replacement are rare events, giving the exploit of anti-correlated profiles a low coverage.

In general, anti-correlation in phylogenetic profiles is also more difficult to detect, because of one additional complication: while the genes may be nicely complementary in their occurrence and not be observed present together, they may very well be absent together. This is because the function they both represent may be entirely dispensable in some genomes, making both genes superfluous. The shared absences may to some extent mask the anti-correlation and usually need to be filtered away by only considering genomes known or suspected to encode the respective function [39].

5 Outlook

5.1 Methods based on Sequence Evolution

Interaction prediction based on genomic context is usually not concerned with the actual nucleotide sequences of the genes in question (or with the amino acid sequences of their encoded proteins). Sequences are only of interest inasmuch as they allow the delineation of orthology relations across genomes. Apart from that, mainly the arrangement and occurrence of the various genes relative to each other are of interest, as well as their presence in certain genome subsets.

However, at least in the case of proteins interacting directly through physical contact, the sequences themselves may also hold valuable information about the interaction. This is because of an effect termed *correlated mutations* [20, 44]. Correlated mutations may arise at the interaction interface between

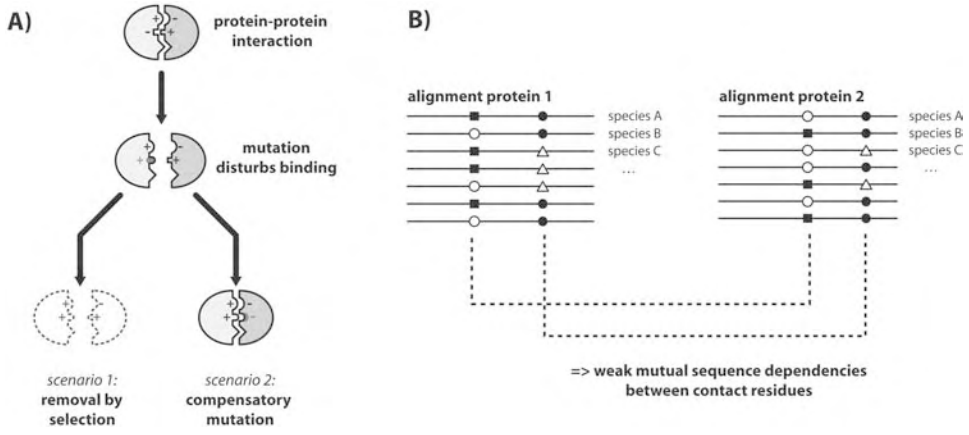

A)
protein-protein interaction
mutation disturbs binding
scenario 1: removal by selection
scenario 2: compensatory mutation
B)
alignment protein 1
alignment protein 2
species A
species B
species C
...
=> weak mutual sequence dependencies between contact residues

Figure 6 Contact interfaces between proteins may lead to dependencies in their amino acid sequences. (A) Possible outcomes of a mutation that compromises/weakens a binding interface. Scenario 1 is by far the most frequent; the mutation is selected against and removed from the population. Scenario 2 happens only rarely; a second mutation, in the other protein, has compensated for the adverse effect of the initial mutation. (B) Multiple events of the latter type may lead to weak dependencies between some alignment columns of the proteins (physicochemical or sterical properties of the residues may have to remain compatible).

two proteins and are basically the result of mutations in one binding partner compensating the effects of mutations in the other binding partner. For example, if one residue at the interaction interface mutates from an amino acid with a small side-chain to an amino acid with a bulky side-chain, this may have negative effects on the strength and specificity of the binding: the bulky side-chain may cause a steric hindrance for the binding. In such a situation, two scenarios are conceivable: either the mutation causes a decrease in fitness, is thus selected against and eventually removed from the population, or it may occur that a second mutation (in the other protein) happens to have a compensatory effect in the binding surface and restores the quality of the binding (Figure 6). The first scenario is much more likely to happen, but even if events of the second type occur only occasionally, they have the potential to cause a weak correlation between the sequences of the two gene families. In principle, this "dependency" in sequence evolution can be detected and exploited for interaction prediction, using tools from information theory.

This approach may roughly qualify as a genomic context approach, because it relies on nothing more than protein sequences available from complete genomes, grouped into families according to orthology relations. It is potentially a very powerful approach: not only does it predict the exact type of a functional association (physical binding), but it may also predict binding interfaces and the topology of protein complexes.

Unfortunately, the correlated mutation signal in protein sequences is quite weak. Apart from correlated mutations, many other factors shape protein sequences: simple drift due to neutral mutations, selective constraints due to folding (folding often happens independently of a binding partner), as well as sequence conservation due to functional requirements such as active sites or substrate binding. The weak signal in correlated mutations means that a large number of sequences need to be analyzed and confounding signals/noise removed before a confident interaction prediction can be made. Currently, the method's power as an interaction prediction tool is still somewhat limited [44]; however, it is bound to improve with more genomes and, just like the other genomic context approaches, it is probably best used in combination with other prediction tools.

5.2 Web-based Implementations of Genomic Context Tools

Genomic context studies involve a considerable amount of work: hundreds of genomes need to be analyzed, orthology relations and gene families identified, and observations tested systematically for their relevance. Fortunately, much of this work recurs repeatedly for any type of genomic context analysis and most of it is well-suited for automation. For this reason, databases have been created that contain the results of many of the necessary computations and comparisons, and that hold large numbers of predicted protein–protein associations. These databases are periodically updated, to accommodate newly sequenced genomes. The predicted interactions may be available for download, and the databases often have web-accessible front-ends allowing the convenient browsing and comparison of prediction results (Table 2).

The last step in any genomic context analysis is the least amenable to automation. The predicted interactions need to be assessed for plausibility; they need to be integrated with expert knowledge and considered in the context of accessory information such as protein localization data or in-house experimental data. This last step needs to be done manually, but even for this task it is advantageous to have access to a database with a large number of predictions: The judgment on relevance is most useful when it is done globally – by ranking all the predictions for an organism according to score or "signal strength", then comparing them to previous knowledge and to reference data sets, and deciding which range of predictions is likely to be reliable. For this type of final judgment, a formal framework does not yet exist: no fully parameterized quantitative model for genome evolution is available, which would enable the estimation of an *a priori* likelihood or relevance of a certain observation. This means that any scoring system for ranking genomic context predictions is necessarily somewhat "*ad hoc*" and can provide only a rough estimate of the prediction accuracy. However, a scoring system is

Table 2 Genomic context databases and web tools (as of July 2005)

Name	Methods				Genomes (July 2005)	Comments	Ref.	URL
	N	F	P	O				
ERGO	√	√	√	√	330	a complex genome annotation and pathway discovery system, includes genomic context analysis modules; *only available commercially*	43	ergo.integratedgenomics.com
FusionDB		√			89	dedicated specifically to gene fusion analysis; for each fused gene, detailed evidence including alignments and phylogenetic trees of the protein sequences is displayed	54	igs-server.cnrs-mrs.fr/FusionDB
GeConT	√				238	focused on neighborhood analysis; compact and informative display of genomic situation; no scoring/ranking of predicted partners	9	www.ibt.unam.mx/biocomputo/gecont.html
Nebulon	√				229	specifically designed to allow "operon walking", i.e. to study the rearrangement of operon fragments in larger functional systems	27	http://tikal.cifn.unam.mx/nebulon
Phydbac	√		√		150	focused on the analysis of phylogenetic profiles; genes are not simply assigned as "present" or "absent", but represented by their score in similarity searches	13	igs-server.cnrs-mrs.fr/phydbac
PLEX	√	√	√		89	main focus is on analysis of phylogenetic profiles; genes are represented by their score in similarity searches; the neighborhood module only inspects one genome at a time	12	bioinformatics.icmb.utexas.edu/plex
Predictome	√	√	√		71	one of the earliest integrated genomic context analysis tools; cooperates with the VISANT network visualization tool; maps GO annotations to network	37	predictome.bu.edu
PROLINKS	√	√	√		163	Integrates genomic context predictions into a network of interactions; individual methods can be turned on and off; includes a text-mining module	8	dip.doe-mbi.ucla.edu/pronav
SEED	√	√		√	314	a comprehensive system for genome and pathway annotation, includes genomic context tools; complex user interface, can be installed and maintained locally by users	41	theseed.uchicago.edu/FIG/index.cgi
STRING	√	√	√	√	179	focused on scoring the predicted interactions and on inter-species knowledge transfer; apart from genomic context tools, includes text mining and imported experimental data, as well as coexpression information	60	string.embl.de

Available analysis methods are abbreviated as follows: N = conserved neighborhood; F = gene fusion analysis; P = phylogenetic profiles (gene co-occurrence analysis); O = other data and methods.

quite essential when dealing with genomic context predictions. A single query gene can have a multitude of predicted partners, especially when analyzing several types of genomic context simultaneously. Without a score providing the ranking of predictions, it can be very difficult to discern false positives from likely true positives.

Another advantage of globally precomputed databases is that they enable the presentation of a large number of predicted interactions in a graphical summary. Often, this is done in the form of an interaction network: nodes in the network represent proteins (or orthologous groups of proteins) and edges in the network represent predicted interactions. This enables the user to discover higher-order dependencies in the data, such as dense clusters of interactions that may represent functional subsystems (modules). Some web tools will also display associated information in the networks: experimental data, previous knowledge and detailed protein features such as their domain composition.

When using web tools and databases for genomic context analysis, it is generally prudent to have a look at the documentation. Each tool will follow its own philosophy and procedures, and results can therefore be quite different. The documentation should allow the user to understand what procedures were applied, and the user can then assess why differences may arise and how they can affect results in a particular area of interest. At the end of a genomic context analysis, the user might even want to follow-up with a manual, in-depth analysis – especially if initial results are promising. This may be needed because the databases and web tools are necessarily never quite up to date; there will usually be more data available in the form of newly sequenced genomes, which can be added to strengthen the analysis.

5.3 Scoring and Integration

As is the case with other bioinformatics prediction techniques, it is crucial in genomic context analysis to provide users with a score for each prediction, so that they can roughly judge its reliability. Scoring systems can take advantage of the fact that each genomic context technique produces a quantitative measure that its correlated with the "signal strength" of the predicted associations: in the case of gene neighborhood, for example, the quantitative measure lies in the number of genomes in which a particular gene neighborhood is found – the more genomes, the stronger the signal. For gene co-occurrence, the signal strength roughly corresponds to the degree of similarity between the phylogenetic profiles and for gene fusion it is the number of genomes that contain a given fusion architecture. To convert signal strength into an estimate of reliability, the predictions are usually benchmarked against a reference set of known functional associations or pathways. Such reference sets are typically based on externally provided expert annotations that group proteins into pathways, complexes, or otherwise similar functional “roles”. Reference sets that are frequently used include the pathway maps of the KEGG database [29], functional classification schemes such as the Gene Ontology [21] or the COGs database [55], or matches in Swiss-Prot keyword assignments [3]. Benchmarking itself is relatively straightforward: the predictions are binned according to genomic context signal strength, and the amount of false-positive and false-negative predictions is assessed for each bin. This leads to statements such as: “a gene neighborhood observed in more than three distinct bacterial phyla has a likelihood of above 95% of correctly indicating a functional link”.

It should be noted that such benchmarked scoring systems still provide only a rough approximation of the actual reliability of a prediction. This is due to the incomplete and possibly biased nature of the various reference sets. For example, a given reference set may be focused on a particular functional area only (e.g. metabolism in the case of KEGG); this may lead to overly confident predictions if a methods works particularly well in that area, but less well in other areas. Furthermore, some classes of genes may be lacking almost entirely from reference sets, for example fast-evolving or rarely occurring genes. Reference sets are often available for a limited number of model organisms only – providing scores for other organisms requires the assumption that a method works equally for these, which may or may not be true.

After scoring each individual prediction, it is often necessary to integrate the scores, e.g. when a particular interaction is supported by more than one type of prediction method or if there is additional experimental information to take into account. This type of integration across prediction methods and data sets is usually done in a probabilistic fashion, using Bayesian statistics [28,35]. In doing so, it is important to assess whether or not the different data sets are independent of each other. When full independence can be assumed, *naïve* Bayesian integration is sufficient – otherwise more sophisticated approaches are needed. It can be difficult to estimate the relative dependence (correlation) between the various data sets, so heuristic approaches are sometimes taken, in order to globally assign the relative weight of multiple evidence (e.g. Ref. [35]).

5.4 Genome Sequencing Strategies: Impact on Genomic Context Analysis

Genomic context techniques, like many other bioinformatics approaches, will become more and more powerful with the advent of further genome sequences [7]. Hopefully, genome sequencing will remain a mainstay of biology for some time to come – each newly completed genome has an intrinsic value that goes far beyond the initial applications foreseen at the time of sequencing. Only complete genomes can provide the molecular basis for a full understanding of organisms. For comparative genomics in particular (with all its applications in evolution and molecular function), a continued flux of genome sequences is essential.

However, given limited resources, there is always the question of how to prioritize sequencing, and at what level of precision and completeness to be satisfied in each individual sequencing project. Currently, there is an increasing trend for genome projects to be specifically designed for low coverage and precision – an approach termed "survey sequencing" (e.g. Refs. [30, 53]). While such projects provide valuable early characterizations of genomes (and give a welcome head start in the race to publication), they also run the danger of postponing indefinitely the necessary finishing work. In fact, many of the prominent animal genomes available today are still not finished, to the extent that some are not more than a large collection of incomplete fragments (e.g. Ref. [2]). To some extent, this de-values the initial investments – by effectively limiting the usefulness of genome data for studies in areas such as genome synteny, gene structure, duplication polymorphisms or gene content.

For genomic context techniques, it is important to be aware of the completion status of the genome sequences used. Phylogenetic profiles, for example, suffer a loss in resolution and predictive power when there are many sequencing gaps – any gene missed in sequencing will have the same effect as a gene loss, i.e. it will appear as an evolutionary event carrying a selective signal. This can lead to high levels of noise in phylogenetic profiles, especially for gene families that are rarely lost biologically, or for organisms in which gene loss constitutes a dominant part of the signal (such as higher eukaryotes). In addition, sequencing gaps can lead to problems when assigning orthology relations between species. Many orthology approaches rely on finding the "most similar" relative of a gene in the other organism, but this can lead to wrong results when the best relative has been missed in sequencing.

Another important issue is the selection of genomes to include in a genomic context analysis; with many more genome sequences expected, it may soon be no longer feasible to include all genomes for an analysis. In that case, it is important to select genomes that cover as widely as possible the known phylogenetic ranges of life. The goal should be to cover as much evolutionary time as possible, in order to maximize the cumulative impact of selection on the sequences studied. Formally, this can be achieved by maximizing the cumulative branch lengths of a whole-genome phylogeny, while at the same time minimizing the number of genomes considered. Effectively, one wishes to select against "redundant" (i.e. closely related) genomes, which do not provide much new signal in genomic context analysis. For comparative genomics in general, it would be desirable if genome sequencing projects were to follow this principle as well, trying to focus on deeply branching, underrepresented phyla. However, sequencing priorities of the scientific community are, necessarily, driven not only by phylogenetic considerations, but also by medical, economical and feasibility questions. This leads to a strong bias in available genomes: some parts of the tree of life are strongly underrepresented (certain bacterial phyla may be difficult to cultivate and do not have a single sequenced representative), whereas others are covered redundantly (of *E. coli* alone, several strains have been sequenced). Fortunately, future sequencing projects seem to be increasingly directed also by evolutionary and phylogenetic considerations [10], leading to a better representation of biodiversity and evolution.

5.5 Environmental Context

Each time a new genome is sequenced, we learn more about the metabolic and cellular capabilities of a particular organism. However, genomes also provide another type of information: they allow inferences about the environments in which the organisms live. For example, if a genome harbors photosynthesis genes, we can assume that the organism lives – at least occasionally – in an environment exposed to light; conversely, if a genome lacks certain genes (e.g. genes needed to produce an essential metabolite), we can assume that the metabolite in question is somehow available from the environment. Thus, genome sequences reflect environmental demands and opportunities – an assumption that is frequently the basis for interpreting novel genome sequences (e.g. Refs. [4,23]).

Taking this a step further, we may not even need fully sequenced genomes to learn something about an environment – instead, sequencing directly DNA that has been isolated from the environment should be almost as informative. Data of this type is increasingly becoming available through so-called *environmental genomics* or *metagenomics* projects, which are an alternative to traditional genome sequencing. In metagenomics, DNA is isolated directly and indiscriminately from an environment of interest (e.g. soil, water or even air), and is subjected to shotgun sequencing – irrespective of the complexity of the species mixture in the sample. Metagenomics data provide a very comprehensive and unbiased view on the coding potential of a biological community in an environment, much better than what can be achieved through traditional genome sequencing (traditional sequencing requires clonal growth of a single isolated organism in the lab, but a surprisingly large fraction of organisms in nature resist cultivation in a laboratory setting).

The increased availability of metagenomics data may soon enable an exciting extension to genomic context techniques – through an approach that might be termed "environmental context". Here, the presence and absence of a particular gene family in various environmental metagenomes (i.e. its "environmental profile") is assumed to depend on the gene's function – and this in turn means that functional partners should have similar environmental profiles. In the future, dedicated databases could hold such information (e.g. environmental presence/absence profiles) for all known gene families – updated with new information each time a novel metagenomics data set becomes available. Systematic searches in this database would then identify sets of genes that have higher-than-random similarities in their profiles and, thus, predict that they function together.

References

1 ADACHI, N. AND M. R. LIEBER. 2002. Bidirectional gene organization: a common architectural feature of the human genome. Cell **109**: 807–9.

2 APARICIO, S., J. CHAPMAN, E. STUPKA, et al. 2002. Whole-genome shotgun assembly and analysis of the genome of *Fugu rubripes*. Science **297**: 1301–10.

3 BAIROCH, A., R. APWEILER, C. H. WU, et al. 2005. The Universal Protein Resource (UniProt). Nucleic Acids Res. **33**: D154–9.

4 BENTLEY, S. D., K. F. CHATER, A. M. CERDENO-TARRAGA, et al. 2002. Complete genome sequence of the model actinomycete *Streptomyces coelicolor* A3(2). Nature **417**: 141–7.

5 BLUMENTHAL, T. 2004. Operons in eukaryotes. Brief. Funct. Genomics Proteomics **3**: 199–211.

6 BOBIK, T. A. AND M. E. RASCHE. 2001. Identification of the human methylmalonyl-CoA racemase gene based on the analysis of prokaryotic gene arrangements. Implications for decoding the human genome. J. Biol. Chem. **276**: 37194–8.

7 BORK, P., L. J. JENSEN, C. VON MERING, A. K. RAMANI, I. LEE AND E. M. MARCOTTE. 2004. Protein interaction networks from yeast to human. Curr. Opin. Struct. Biol. **14**: 292–9.

8 BOWERS, P. M., M. PELLEGRINI, M. J. THOMPSON, J. FIERRO, T. O. YEATES AND D. EISENBERG. 2004. Prolinks: a database of protein functional linkages derived from coevolution. Genome Biol. **5**: R35.

9 CIRIA, R., C. ABREU-GOODGER, E. MORETT AND E. MERINO. 2004. GeConT: gene context analysis. Bioinformatics **20**: 2307–8.

10 COUZIN, J. 2003. Genomics. Sequencers examine priorities. Science **301**: 1176–7.

11 DANDEKAR, T., B. SNEL, M. HUYNEN AND P. BORK. 1998. Conservation of gene order: a fingerprint of proteins that physically interact. Trends Biochem. Sci. **23**: 324–8.

12 DATE, S. V. AND E. M. MARCOTTE. 2005. Protein function prediction using the Protein Link EXplorer (PLEX). Bioinformatics **21**: 2558–9.

13 ENAULT, F., K. SUHRE, O. POIROT, C. ABERGEL AND J. M. CLAVERIE. 2004. Phydbac2: improved inference of gene function using interactive phylogenomic profiling and chromosomal location analysis. Nucleic Acids Res. **32**: W336–9.

14 ENRIGHT, A. J., I. ILIOPOULOS, N. C. KYRPIDES AND C. A. OUZOUNIS. 1999. Protein interaction maps for complete genomes based on gene fusion events. Nature **402**: 86–90.

15 FITCH, W. M. 1970. Distinguishing homologous from analogous proteins. Syst. Zool. **19**: 99–113.

16 Fitch, W. M. 2000. Homology a personal view on some of the problems. Trends Genet. **16**: 227–31.

17 Fraser, C. M., J. D. Gocayne, et al. 1995. The minimal gene complement of *Mycoplasma genitalium*. Science **270**: 397–403.

18 Fukuchi, S. and K. Nishikawa. 2004. Estimation of the number of authentic orphan genes in bacterial genomes. DNA Res. **11**: 219–31, 311–3.

19 Galperin, M. Y. and E. V. Koonin. 2000. Who's your neighbor? New computational approaches for functional genomics. Nat. Biotechnol. **18**: 609–13.

20 Gobel, U., C. Sander, R. Schneider and A. Valencia. 1994. Correlated mutations and residue contacts in proteins. Proteins **18**: 309–17.

21 Harris, M. A., J. Clark, A. Ireland, et al. 2004. The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res. **32**: D258–61.

22 Heath, R. J. and C. O. Rock. 2000. A triclosan-resistant bacterial enzyme. Nature **406**: 145–6.

23 Hess, W. R. 2004. Genome analysis of marine photosynthetic microbes and their global role. Curr. Opin. Biotechnol. **15**: 191–8.

24 Hillier, L. W., W. Miller, E. Birney, et al. 2004. Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. Nature **432**: 695–716.

25 Huynen, M. A. and P. Bork. 1998. Measuring genome evolution. Proc. Natl Acad. Sci. USA **95**: 5849–56.

26 Jacob, F. and J. Monod. 1961. Genetic regulatory mechanisms in the synthesis of proteins. J. Mol. Biol. **3**: 318–56.

27 Janga, S. C., J. Collado-Vides and G. Moreno-Hagelsieb. 2005. Nebulon: a system for the inference of functional relationships of gene products from the rearrangement of predicted operons. Nucleic Acids Res. **33**: 2521–30.

28 Jansen, R., H. Yu, D. Greenbaum, Y. Kluger, et al. 2003. A Bayesian networks approach for predicting protein–protein interactions from genomic data. Science **302**: 449–53.

29 Kanehisa, M., S. Goto, S. Kawashima, Y. Okuno and M. Hattori. 2004. The KEGG resource for deciphering the genome. Nucleic Acids Res. **32**: D277–80.

30 Kirkness, E. F., V. Bafna, A. L. Halpern, et al. 2003. The dog genome: survey sequencing and comparative analysis. Science **301**: 1898–903.

31 Korbel, J. O., L. J. Jensen, C. von Mering and P. Bork. 2004. Analysis of genomic context: prediction of functional associations from conserved bidirectionally transcribed gene pairs. Nat. Biotechnol. **22**: 911–7.

32 Korbel, J. O., B. Snel, M. A. Huynen and P. Bork. 2002. SHOT: a web server for the construction of genome phylogenies. Trends Genet. **18**: 158–62.

33 Kummerfeld, S. K. and S. A. Teichmann. 2005. Relative rates of gene fusion and fission in multi-domain proteins. Trends Genet. **21**: 25–30.

34 Lathe, W. C., 3rd, B. Snel and P. Bork. 2000. Gene context conservation of a higher order than operons. Trends Biochem. Sci. **25**: 474–9.

35 Lee, I., S. V. Date, A. T. Adai and E. M. Marcotte. 2004. A probabilistic functional network of yeast genes. Science **306**: 1555–8.

36 Marcotte, E. M., M. Pellegrini, H. L. Ng, D. W. Rice, T. O. Yeates and D. Eisenberg. 1999. Detecting protein function and protein–protein interactions from genome sequences. Science **285**: 751–3.

37 Mellor, J. C., I. Yanai, K. H. Clodfelter, J. Mintseris and C. DeLisi. 2002. Predictome: a database of putative functional links between proteins. Nucleic Acids Res. **30**: 306–9.

38 Moreno-Hagelsieb, G. and J. Collado-Vides. 2002. A powerful non-homology method for the prediction of operons in prokaryotes. Bioinformatics **18 (Suppl. 1)**: S329–36.

39 Morett, E., J. O. Korbel, E. Rajan, et al. 2003. Systematic discovery of analogous enzymes in thiamin biosynthesis. Nat. Biotechnol. **21**: 790–5.

40 O'Brien, K. P., M. Remm and E. L. Sonnhammer. 2005. Inparanoid: a comprehensive database of eukaryotic orthologs. Nucleic Acids Res. **33**: D476–80.

41 Overbeek, R. 2005. The SEED – an annotation/analysis tool provided by the Fellowship for Interpretation of Genomes. http://theseed.uchicago.edu/FIG/index.cgi.

42 Overbeek, R., M. Fonstein, M. D'Souza, G. D. Pusch and N. Maltsev. 1999. The use of gene clusters to infer functional coupling. Proc. Natl Acad. Sci. USA **96**: 2896–901.

43 Overbeek, R., N. Larsen, T. Walunas, et al. 2003. The ERGO genome analysis and discovery system. Nucleic Acids Res. **31**: 164–71.

44 Pazos, F. and A. Valencia. 2002. *In silico* two-hybrid system for the selection of physically interacting protein pairs. Proteins **47**: 219–27.

45 Pellegrini, M., E. M. Marcotte, M. J. Thompson, D. Eisenberg and T. O. Yeates. 1999. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. Proc. Natl Acad. Sci. USA **96**: 4285–8.

46 Ramani, A. K. and E. M. Marcotte. 2003. Exploiting the co-evolution of interacting proteins to discover interaction specificity. J. Mol. Biol. **327**: 273–84.

47 Raoult, D., S. Audic, C. Robert, et al. 2004. The 1.2-megabase genome sequence of Mimivirus. Science **306**: 1344–50.

48 Rhee, K. Y., M. Opel, E. Ito, S. Hung, S. M. Arfin and G. W. Hatfield. 1999. Transcriptional coupling between the divergent promoters of a prototypic LysR-type regulatory system, the ilvYC operon of *Escherichia coli*. Proc. Natl Acad. Sci. USA **96**: 14294–9.

49 Rocha, E. P. 2004. The replication-related organization of bacterial genomes. Microbiology **150**: 1609–27.

50 Rost, B. 2002. Enzyme function less conserved than anticipated. J. Mol. Biol. **318**: 595–608.

51 Snel, B., P. Bork and M. Huynen. 2000. Genome evolution. Gene fusion versus gene fission. Trends Genet. **16**: 9–11.

52 Snel, B., M. A. Huynen and B. E. Dutilh. 2005. Genome trees and the nature of genome evolution. Annu. Rev. Microbiol. **59**: 191–209.

53 Strong, W. B. and R. G. Nelson. 2000. Preliminary profile of the *Cryptosporidium parvum* genome: an expressed sequence tag and genome survey sequence analysis. Mol. Biochem. Parasitol. **107**: 1–32.

54 Suhre, K. and J. M. Claverie. 2004. FusionDB: a database for in-depth analysis of prokaryotic gene fusion events. Nucleic Acids Res. **32**: D273–6.

55 Tatusov, R. L., N. D. Fedorova, J. D. Jackson, et al. 2003. The COG database: an updated version includes eukaryotes. BMC Bioinformatics **4**: 41.

56 Tian, W. and J. Skolnick. 2003. How well is enzyme function conserved as a function of pairwise sequence identity? J. Mol. Biol. **333**: 863–82.

57 Tsoka, S. and C. A. Ouzounis. 2000. Prediction of protein interactions: metabolic enzymes are frequently involved in gene fusion. Nat. Genet. **26**: 141–2.

58 von Mering, C. and P. Bork. 2002. Teamed up for transcription. Nature **417**: 797–8.

59 von Mering, C., M. Huynen, D. Jaeggi, S. Schmidt, P. Bork and B. Snel. 2003. STRING: a database of predicted functional associations between proteins. Nucleic Acids Res. **31**: 258–61.

60 von Mering, C., L. J. Jensen, B. Snel, et al. 2005. STRING: known and predicted protein–protein associations, integrated and transferred across organisms. Nucleic Acids Res. **33**: D433–7.

61 Welch, G. R. and J. S. Easterby. 1994. Metabolic channeling versus free diffusion: transition-time analysis. Trends Biochem. Sci. **19**: 193–7.

33 Inferring Protein Function from Protein Structure

Francisco S. Domingues and Thomas Lengauer

1 Introduction

It is well known that structure dictates function in many aspects of reality. This observation holds for the machines and buildings that we engineer. It also holds for the anatomy and physiology of the diverse types of organisms studied in biology, as well as to the molecular biology underlying the living processes of these organisms. Nevertheless, the determination of the protein function given its structure is not a trivial problem. The current chapter presents the recent developments in this field of structural bioinformatics.

So far, most of the investigation on the relationships between the structure and function of proteins has centred on the analysis of single case experimental results. Only recently have there been attempts to develop computational methods in order to find general rules behind these relationships and to generate predictions and testable hypotheses. There are two challenges at the center of these efforts: the localization of functional sites in proteins, and the identification of the molecular function of a protein. A considerable number of approaches have been proposed recently to address these challenges. Some of these approaches are still very new and untested, but others are more mature and have been successfully applied in the functional characterization of several proteins. This new field is now making its first contributions to the understanding of living processes at the molecular level, a necessary step along the path from genomes to therapies.

Structural models provide vital information as we seek to comprehend how proteins function at the molecular level. In this section we first address the different notions of protein function and the different kinds of functional information that is provided by the structure of a protein. We then focus on the structure–function relationships.

Bioinformatics - From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

1.1 Different Levels of Protein Function

Function is not a simple concept. For a comprehensive definition, several levels have to be taken into account [13,97] (see also Chapter 29). The molecular or biochemical function pertains to the chemical interactions and reactions in which the protein participates. Function can also be defined on a broader level in terms of the biological processes in which the protein is involved, including its cellular and physiological roles.

In general, proteins bind to other molecules in order to perform their function. Therefore, binding can be regarded as a fundamental molecular function [97]. For example, proteins bind to other proteins to regulate their function or to transmit information along signal cascades, they bind to DNA in order to control gene expression, enzymes bind to the respective substrates as well as to cofactors and effectors in order to catalyze certain reactions, and transporters and receptors bind to the respective ligands in order to afford molecular transport or cellular communication.

1.2 Structural Models

Structural models can be determined experimentally by X-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy or predicted using different computational approaches (see Chapters 10–12). Experimental models are much more reliable than predicted models, but in general the experimental approach is more costly and time consuming. Fortunately, over the past years there have been significant improvements both in the automation of experimental structure determination and in the quality of the predicted models.

Structural models provide the three-dimensional (3-D) coordinates of the protein atoms. From these coordinates it is possible to characterize the protein in terms of geometry and chemistry. In particular, it is possible to identify the secondary structure elements and the organization of the polypeptide chain in 3-D (*fold*). Surface and buried atoms can also be identified, as well as the overall protein shape and surface chemistry. It is also possible to analyze the local arrangement of the atoms in any part of the model and characterize their local chemical environments. Many experimental models include several polypetide chains – from these models it is possible to identify and characterize the protein–protein interactions. Other models include small molecules bound to the proteins, either as natural substrates/ligands, or as artificial inhibitors or modified ligands. These protein–ligand complexes provide information regarding the location of the functional sites, the nature of the protein–ligand interactions and possibly even the molecular mechanisms underlying the protein function. Obviously, these structural models provide valuable information regarding molecular function [120]. Nevertheless, it is not trivial, at least with current methods, to infer the biological, cellular or physiological role of a protein based on structure data.

1.3 Homology and Function

Searching for homology has been a predominant approach in bioinformatics. This is understandable given that sequences are easily determined, and that the resulting data are suited quite well for a wide range of computational and statistical approaches for homology detection. This has produced important evolutionary insights and also provided functional information indirectly (see Chapters 3, 4, 11, 30 and 37).

An evolutionary relationship between two proteins indicates a possible function relationship, where the confidence for functional relationship increases with the percentage of sequence identity. It has been found that function is conserved between two enzymes when they share more than 40% of sequence identity (close homology) [121, 134]. Structure is more conserved in evolution than sequence. Therefore, methods for backbone structure comparison have been applied to detect remote homology between proteins, where the sequence similarity is not significant. Nevertheless, fold similarity does not necessarily imply homology, as proteins can share similar folds in the absence of homology. Backbone structure comparison methods are usually combined with other approaches (including functional information) in order to predict remote homology [35]. Functional variation is significant in the case of remote homology [121, 134], therefore inferring function relationship between remote homologous proteins is even more challenging than in the case of close homology. In addition, proteins can share similar molecular function in the absence of homology [42].

Despite these difficulties, methods for the comparison of protein structures in terms of backbone geometries are now routinely used in the functional characterization of newly determined structures. The process is not automated and relies extensively on human expertise for the interpretation of results. Usually a query structure is compared to all the structures in the Protein Data Bank (PDB) [16], and the entries with the most similar backbone structure to the query are identified. In general, most similar structures correspond to the homologous proteins (if any). Most of these methods also generate a structural alignment between the two proteins compared. Detailed analysis of this structure alignment, particularly of the aligned functional sites, is of great value for inferring functional relationships. The success of this approach is demonstrated by the considerable list of proteins functionally characterized using backbone structural comparison [136]. Severall backbone structure comparison methods are available and have been recently reviewed [66, 111].

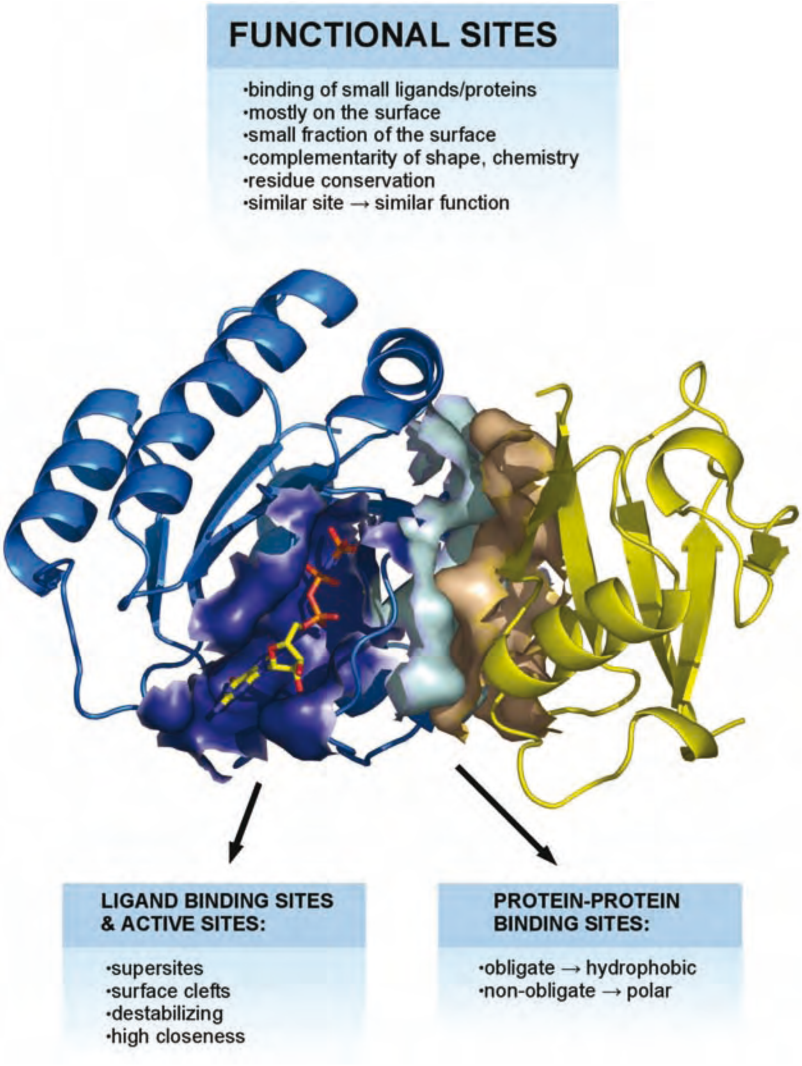

FUNCTIONAL SITES
•binding of small ligands/proteins
•mostly on the surface
•small fraction of the surface
•complementarity of shape, chemistry
•residue conservation
•similar site → similar function
LIGAND BINDING SITES & ACTIVE SITES:
•supersites
•surface clefts
•destabilizing
•high closeness
PROTEIN-PROTEIN BINDING SITES:
•obligate → hydrophobic
•non-obligate → polar

Figure 1 General properties of functional sites. Complex of Ras (blue) with RalGDS (yellow), PDB code 1lfd [51]. Protein–protein binding sites in light blue and orange. Substrate-binding site in dark blue, with bound substrate analog. Graphics produced with PyMOL (http://www.pymol.org).

1.4 Structure and Function

Functional sites share geometric and physicochemical properties [97], (Figure 1). These properties provide the basis for different computational methods for function prediction.

As mentioned before, binding is a common theme in protein function. Binding usually takes place in a localized region at the protein surface, which typically corresponds to a relatively small fraction of the total surface. It consists of several key chemical groups from some residues that cluster at the protein surface independent of the position along the sequence. The rest of the protein serves as a structural framework that provides both the stability and flexibility necessary for shaping the functional region. Complementarity of shape and chemistry contributes to specific binding (lock-and-key model). However, proteins are flexible, so this complementarity is not always evident before binding, as proteins can adjust their conformation upon ligand binding (induced-fit model).

In general, the binding sites at the protein–protein interfaces display a high degree of complementarity with respect to shape and chemistry. This complementarity is achieved with different shapes, varying from flat surfaces to interfaces consisting of a protruding surface binding a cavity. Different physicochemical environments are also observed in different types of protein–protein interactions [86]. Hydrophobic residues tend to cluster at the interfaces of obligate interactions. These obligate interactions are required for the stability of the proteins that form the complex. In non-obligate interactions, the participating proteins can fold and form stable structures independently. These interfaces tend to include polar residues.

If the ligand is a small molecule, the interactions are usually located at surface clefts. These clefts provide easy accessibility for the ligand. In addition, high complementarity and increased binding specificity can be achieved by enveloping the ligand. These are general rules, but it is not difficult to find exceptions – a common theme in biological systems. For example, some ligands are found deeply buried inside the protein, requiring large conformational changes for the ligand to move in and out.

Functional sites are subjected to selective pressure different from the rest of the protein surface, as their structural and chemical integrity is required in order to preserve molecular function. In particular, functional-site residues are less tolerant to mutations than the rest of the surface residues. If the molecular function is preserved in two homologous proteins, the functional sites tend to display higher sequence conservation than the rest of the protein surface. Functional sites have also been found to be destabilizing (energetically unfavorable) [107,108]. The rest of the protein structure has to compensate for this effect in order to guarantee overall stability. In addition, analysis of residue interaction graphs, where protein residues are the nodes and interactions are edges, has revealed that active-site residues tend to have a relatively low average distance to all other residues in the graph (high closeness) [3].

Functional sites of different proteins exhibiting similar molecular function have also been found to share conserved geometry and chemistry. This property is quite relevant for the prediction of protein function. If two proteins have similar functional sites, it is likely that they have similar molecular function. This is valid independently of the overall fold or sequence similarity, or whether the proteins are homologous or not. The conserved ATP recognition in different types of enzymes is one such example [34]. Defining function similarity in this respect is not trivial. For example, enzyme similarity can be observed at the level of the substrate or cofactor binding site, the catalytic site, or the mechanism of catalysis.

Finally, it has been observed that the locations of small-molecule ligand-binding sites are conserved among some protein folds [101]. This conservation of binding sites can be observed even between nonhomologous proteins with the same fold.

1.5 Why Predict Function from Structure

The current motivation for function prediction based on structure information involves several factors. (i) The increasing availability of structural data results in many opportunities for computational analysis. (ii) The many hypothetical proteins whose structures are becoming available via structural genomics projects need to be characterized functionally. (iii) Improving the understanding of protein function based on available structural information is expected to generate deep biological insight. (iv) A rising interest in more direct approaches for characterizing proteins functionally.

Over the past 20 years the improvements in the techniques for protein structure determination have resulted in increasing numbers of experimental models being made publicly available in the PDB [16]. The amount of structural data achieved a certain level of critical mass, in the sense that it is now possible to train and test statistical learning and computational methods for characterizing structures and generating predictions in an automated fashion. Knowledge-based approaches and statistical learning methods are at the center of the new field of structural bioinformatics.

The current structural genomic initiatives (Chapter 13) are another factor. As an outcome of these initiatives we now have many structures available for proteins with uncharacterized function and many more are to be expected. This creates a demand for computational tools for functional characterization that make use of the available structural information.

We should not assume that these tools are only applicable to the structural models of hypothetical proteins. When applied to already annotated proteins, such methods can provide additional functional information. In particular, many proteins are known to have multiple functions [32, 55]. It is reasonable to think that this is the case for many more proteins which are yet to be identified. New and exciting functional insights can be expected, as these tools are applied to structures of partially characterized proteins.

Finally, one can observe a growing interest in more direct approaches for function characterization than the traditional homology detection methods. An important development has been Gene Ontology (GO), a systematic ontology for gene function [9]. An example of this new focus on functional characterization was the development of genomic context methods to predict which proteins participate in common biological processes. These methods are based on the comparison between genomes of different species (see Chapter 32). Approaches to functionally characterize proteins based on structural information are also part of this trend.

1.6 The Challenges of Automatic Prediction of Function from Structure

The two main challenges in prediction of function from structure are the localization of functional sites and the characterization of molecular function. Different approaches have been proposed in order to locate functional sites: structure comparison, quantification of structural features, and a combination of structural information and evolutionary conservation. They will be described in the Section 2. Characterization of molecular function is the other great challenge and is described in Section 3. The field is still in a highly experimental stage, and many different approaches are being proposed and tested. Among these alternative approaches we find several based on local structure comparison. Their goal is to find similar local structures (*motifs*) in different proteins. If these structural motifs correspond to functional sites, the functional information can be transferred between the matching proteins independent of the evolutionary relationships. An important element in the search for structural motifs is the availability of a database of functional sites.

1.7 Structure of the Chapter

It is possible to describe bioinformatics approaches from either the methodological or the application point of view. In this chapter we try to cover both. The two main sections of the chapter address each of the two main challenges (application): site localization (Section 2) and function characterization (Section 3). Each of these sections is subdivided according to the different approaches (*methodology*). We then mention preliminary efforts to address both challenges (Section 4). We also provide a list with the main tools currently available (Section 5) and apply them to a concrete example. We then describe some of recent successes in structure-based functional characterization (Section 6). We complete the chapter with a look at the future developments (Section 7).

2 Localization of Functional Sites

As described above (Section 1.4), functional sites have several unique features. These are explored by different approaches to locating functional sites in the protein structure.

2.1 Supersites

Structure comparison and structure classification are well-established fields of structural bioinformatics. Superfolds correspond to frequently observed folds which are shared by different nonhomologous proteins [89]. Russell and coworkers have investigated the localization of ligand-binding sites among proteins with the same fold [101]. They have identified supersites, which correspond to conserved structural regions within different superfolds with a significant tendency to bind ligands. Different supersites have been identified, each associated with a different superfold. If the fold of a given protein corresponds to one of these superfolds, the ligand-binding site is likely to match the corresponding supersite.

2.2 Electrostatics

Functional sites are quite unique in terms of local structure [96] – they are generally strained and destabilizing. This strain can become apparent in theoretical estimates of stability based on continuum electrostatics calculations. Two methods have been proposed based on this principle. One of the methods identifies the location of general functional sites [37], while the other focuses on the identification of active sites of enzymes [15]. A related approach uses computed titration curves for the same purpose [64, 88].

Nucleic acid-binding sites are a special case of protein functional sites, in which charge complementarity plays an important role. Two methods have been proposed for the identification of nucleic acid-binding sites. They are based on the identification of positively charged patches with electrostatic potentials [57, 118].

2.3 Surface Geometry

The shape of a site binding a small ligand has unique features in comparison to the remaining protein surface. It has been observed that ligands tend to bind to the largest surface cleft [70]. The SURFNET program locates protein surface clefts or pockets and computes their volume [74]. SURFNET can be used to locate functional sites by identification of the largest surface clefts. To locate these surface clefts, "gap spheres" of a certain radius (4 Å) are placed midway between each pair of atoms and the size of the sphere is reduced until all clashes to other nearby atoms are removed. The sphere is retained if the radius is larger than 1 Å. A contour surface is generated around the clusters of overlapping spheres, which represent the surface clefts and cavities in the protein. The volume of each surface cleft is calculated and the cleft with largest volume is predicted to be the ligand binding site.

Other approaches rely on the α shape algorithm to detect protein surface clefts. The α shape is derived from the weighted Delaunay triangulation of the point coodinates of the atom centers, where Voronoi edges and vertices outside the protein define empty tetrahedra and are excluded. In the APROPOS [95] method, α shapes are used to describe the protein surface at two different resolutions. The higher-resolution α shape includes surface details like cavities and the lower resolution just provides the global shape of the protein. The surface pockets or clefts are identified by determining the regions with the largest differences between the two representations. A single α shape representation is used in the CASTp [75] method and pockets are identified from the collection of empty tetrahedra. CASTp is an analytic method and, in comparison to other methods, has the advantage that it does not depend on discretization, iterations or any heuristic parameters. Nevertheless, unlike SURFNET or APROPOS, CASTp cannot identify clefts corresponding to shallow depressions where the openings are wider than any cross section of the interior. CASTp is available as a web service [20]

SURFNET, APROPOS and CASTp are not the only methods for finding surface clefts. Other methods such as LIGSITE [49] and PASS [25] have been proposed.

2.4 Structure and Evolutionary Information

Several methods have been proposed for the localization of functional sites based on the principle that these sites are subject to a different selective pressure than the rest of the protein along the evolutionary process. The approach has its origins in methods based purely on sequence [31]. More recently, several methods have been proposed that combine evolutionary and structural information in different ways.

2.4.1 Evolutionary Trace (ET)

The ET method was one of the first approaches combining evolutionary and structural information for the purpose of localizing functional regions in proteins [78]. The method is based on several hypotheses. (i) If function is conserved between an ancestral and its descendant proteins, the respective functional sites retain their localization. (ii) If the function is conserved within a subgroup of homologous proteins, the sequence of that functional site will be also be conserved in that subgroup. (iii) If the functions diverge between two subgroups of homologous proteins, one can expect that the functional divergence arises mostly from substitutions at the functional sites. (iv) The functional residues (binding or catalytic sites) are expected to cluster, forming patches along the protein surface.

The ET method consists of several steps. First, given a query protein with known structure, a set of homologous sequences is collected by pairwise sequence comparison. Then, a multiple sequence alignment is generated and a corresponding phylogenetic tree is constructed. The tree is used as a guide to partitioning the set into subgroups of homologous proteins at a certain evolutionary distance. Different partitions can be made at different evolutionary distances. For a given partition, the consensus sequences are determined for each subgroup, describing conserved and nonconserved positions. The evolutionary trace of the entire partition is obtained by aligning the consensus sequences. A position in the trace is neutral if it is nonconserved in any of the consensus sequences. The position is conserved in the trace if it is invariant over all consensus sequences. The position can also be class specific in the trace if it varies between the different subgroups, but is constant within each subgroup. The rank of a residue is the minimum number of subgroups into which the tree has to be divided in order for a residue to be class specific. Residues with low rank number are conserved over most of the tree and residues with high rank number tend to vary even between the most related proteins. Finally, the residues can be mapped into the structure and labeled according to the trace: neutral, conserved or class specific. Functional sites are expected to correspond to conserved and class specific trace residues clustering on the surface of the protein.

Some improvements have been proposed more recently in order to facilitate the use of the method at a large scale. In particular, the method now implicitly takes gaps into account, and calculates a statistical estimate of the significance of the spatial clusters of trace predictions [79, 137]. The method has been applied successfully to the regulators of G-protein signaling (RGS) family of proteins (see Section 6).

2.4.2 **ConSurf**

Several approaches related to the ET method have been proposed. The ConSurf [7,68,100] method tries to improve over ET by weighting amino acid substitutions according to differences in physicochemical properties and by calculating residue conservation scores based on maximum likelihood or empirical Bayesian algorithms. These residue conservation scores are then mapped onto the corresponding structural models. In general, the functional sites correspond to surface patches with high conservation scores (Figure 2). The initial implementation did not provide an objective criterion for defining

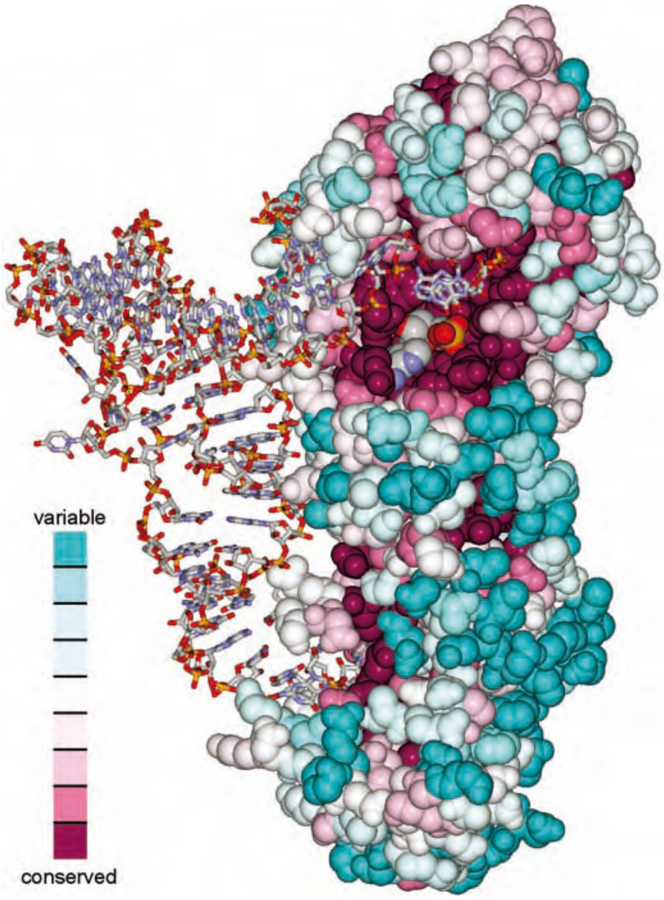


Figure 2 ConSurf results for glutaminyl-tRNA synthetase from *Escherichia coli* complexed with tRNA, PDB code 1gts [94]. Most conserved surface regions match the activation site (top) and the anticodon recognition site (bottom). Graphics produced with RasMol [17, 104] based on a script generated by ConSurf.

the functional region. The authors have now implemented a complementary method (PatchFinder) to automatically identify functional regions [85]. The predicted functional sites correspond to statistically significant clusters of conserved surface residues.

2.4.3 Residue Conservation and Structural Information

Other methods are based on more simplistic models for the conservation of residues while making direct use of structural information [1,22,69]. Sternberg and coworkers proposed a method based on the identification of C^{β} spatial clusters of conserved polar residues [1]. The predicted functional site residues are defined within a sphere centred at the geometric centroid of the conserved spatial cluster. The radius of the sphere is the distance between the centroid and the most distant C^{β} atom of the clustered residues. The approach is only applicable to functional sites with conserved polar residues. This is expected to be the case in most enzyme active sites and in many protein–protein interaction sites.

Landgraf and coworkers have proposed a method for locating functional sites based on two principles [69]. (i) Between functionally conserved proteins the residue conservation of the structural neighbors of a residue at a functional site is larger than the global residue conservation. This is measured by the regional conservation score of a residue. (ii) The global sequence similarities between functionally conserved proteins changes significantly when the same type of analysis is restricted to the structural neighbors of functional site residues. The authors proposed a so-called similarity deviation score of a residue to measure this effect. These scores are calculated from a multiple sequence alignment that includes a protein with known structure and a number of corresponding homologous sequences. From this alignment, two matrices are calculated – the global similarity matrix and the residue-specific regional similarity matrix. The first matrix consists of the pairwise sequence similarity scores between all homologous proteins. The regional similarity matrix includes the same of type of data as the global similarity matrix, but is restricted to the similarity scores of the structural neighbors of the residue. The regional conservation score of a residue amounts to the magnitude of the differences between the two matrices. The similarity deviation score is the correlation between the two matrices.

2.5 Network Centrality

Protein structures have been represented as networks, where nodes correspond to amino acid residues and edges represent interactions. Each residue in the protein can then be characterized in terms of network centrality (closeness centrality), as measured by the inverse of the mean geodesic distance (shortest path) between the node and all other nodes. Amitai and coworkers have shown that enzyme active-site residues tend to have high closeness values [3], which means such residues interact with the other residues directly or by relatively few intermediates. Based on this observation, they proposed SARIG (Structural Analysis of Residue Interaction Graphs) – a method to predict active site residues using closeness and relative solvent accessability. A SARIG web server has been implemented to predict the active site residues and to visualize the distribution of closeness values on the protein structure.

2.6 Combined Approaches

Several methods combine these different approaches in order to locate functional sites. In particular, combined methods have been proposed to identify active sites of enzymes and protein–protein interaction sites.

2.6.1 Catalytic Sites in Enzymes

Catalytic residues constitute a special group of functional residues. They are generally conserved, they tend to be found in the largest clefts, but at the same time, they tend to be buried (low relative solvent accessibility). They tend to be located in coil regions and usually are charged or polar [14]. Based on these observations, Gutteridge and coworkers devised a method for locating active site residues [46]. The following parameters were used to train a neural network: residue conservation, solvent accessibility, localization on a surface cleft and relative size of the cleft, type of secondary structure, and residue type. The neural network score obtained for each residue is used to define a list of possible active-site residues. Then, spatial clustering is performed and the cluster with the best scoring residues is predicted to be the catalytic site. Ota and coworkers addressed the same problem with a related method, based on identification of conserved residues, estimation of destabilization effect and location (cleft versus surface or buried) [90].

2.6.2 Protein–protein Interactions

Protein–protein interactions can be classified into different types, regarding composition (homo-oligomers, hetero-oligomers), autonomy of the protomers (obligate, non-obligate) or lifetime (permanent, transient) [86]. Protein–protein binding interfaces have been characterized regarding different properties: solvation potential (preference for being buried or exposed to solvent), surface residue propensity, hydrophobicity or protrusion. Different trends were observed for the different types of interactions (homodimer, large or small heteroligomers, or antibody–antigens). These observations were the basis for devising different scoring functions for the identification of certain types of interactions [58, 59]. More recently, Neuvirth and coworkers [84] tested the combination of different measures in order to predict transient protein–protein binding sites. The authors found that the most successful combination includes secondary structure, atom distribution, patterns of neighboring amino acids, residue conservation, chemical character of surface atoms (charge, aromaticity, hydrophobicity), position of water in crystal structures, sequence distance and hydrophobic patches. This problem has now been addressed in the PPI-Pred method, using statistical learning approaches in order to find the best combination of properties for each type of interaction [23].

A related problem is to distinguish the native protein–protein interactions (so-called biological interactions) from crystal contacts in structural models determined by X-ray crystallography. The latter contacts are a result of the crystallization process and do not occur natively in solution. This problem has been addressed in the Protein Quaternary Structure database (PQS) [50]. PQS is based on a method that empirically combines different interface physical measures, particularly the contact area, in order to discriminate biological from nonbiological contacts. It provides as a result the predicted quarternary state for each PDB entry. Other approaches have combined interaction size and residue conservation with a neural network predictor [124]. More recently, the NOXclass classification method has been proposed (NOXclass: non-obligate, obligate and crystal-packing classification). Different interface features have been combined in NOXclass with a support vector machine (SVM) implementation to classify a given protein–protein interface into crystal contacts, non-obligate or obligate interactions [140].

3 Characterization of Molecular Function

Characterization of molecular function based on protein structure is a considerable challenge. Direct comparison of functional sites provides a more direct approach to function prediction than methods based on homology detection (Section 1.3). The general strategy is to identify similarities between a query, an uncharacterized functional site and a functional site corresponding to a protein of characterized molecular function. The similarities are measured in terms of geometry or physicochemical environment. Based on this similarity, an attempt is made to infer aspects of the molecular function of the query protein.

We start this section by presenting some general concepts regarding functional characterization based on structure and then we describe several methods. First, we focus on the approaches based on comparison of atom coordinates, and then we describe methods that take into account surface shape and the local chemical environment. Finally, we briefly mention different databases of functional sites.

3.1 General Principles

3.1.1 Homology versus Nonhomology

There are different application scenarios in the prediction of function based on structure. A possible application is the identification of homologous proteins with similar function and, therefore, with conserved functional site. As described in Section 1.3, homology can be identified by sequence-based methods and by backbone structure comparison. The sequence-based methods have been combined with rule-based approaches for automated functional annotation of uncharacterized proteins [21, 41]. In addition, an automated method for functional prediction of homologous proteins has been proposed, which combines measures of backbone structural similarity and sequence conservation in the structural alignment [98]. Sequence-based methods lack information regarding the geometry and the distances between the functional groups, and methods based on backbone structural comparison do not center on the functional site. In contrast, methods for function prediction based on local structure similarity take into consideration the particular chemistry and geometry of the functional sites. One can expect that applying such methods to compare the functional sites of homologous proteins will further improve the quality of the functional annotations.

In another application scenario, the goal is to identify common function independent of homology. It is known for some cases that function and functional sites can be conserved regarding chemistry and geometry even in the absence of homology. A classical example is the Ser–His–Asp catalytic triad, found in peptidases with different folds (trypsin-like and subtilisins) [127]. Another example is provided by the similar ATP-binding sites in different ATP-dependent enzymes [34]. The comparison of functional sites can help in the characterization of molecular function, even if the query protein and characterized protein are not homologous.

3.1.2 Uncertainty and Flexibility in the Structural Models

When comparing functional sites one has to take into account that similar molecular functions can correspond to similar functional sites, but one can only expect similarity, not exact conservation of geometries and chemistry. In this respect, uncertainties in the coordinate values should be taken into account first. They correspond to atom vibrations, disorder in the crystal or errors in the structure determination process. Experimental methods always have a certain degree of uncertainty associated with the resulting coordinates, although the actual error estimates might not always be easy to obtain. The resolution and the free R factor (R_{free}) are common indicators for the quality of the crystallographic models. The RMSD across the ensemble of solutions has been used as a measure of quality for NMR structures, but other measures have also been proposed [73]. The quality of predicted structural models has been improving over time, but they still cannot compete with experimental models.

Apart from the flexibility resulting from atom vibrations, proteins are also flexible as a result of collective motions of atoms and residues. One can often observe considerable differences between backbone conformations in alternative structural models of the same protein [36]. Often, the changes comprise the functional sites, as structural changes are generally associated with function. Binding of a ligand, nucleic acid or protein is often associated with structural changes in the functional site.

Ultimately, one can always expect some degree of dissimilarity between different proteins with similar functional sites and molecular function. The challenge is to define a similarity measure and a threshold that can be used to identify the sites with similar molecular function.

3.1.3 **Functional Descriptors, Comparison and Scoring**

Within the set of known protein structures, one can observe conserved regions regarding structure and chemical environment. These recurrent structural/chemical environments correspond to structural or spatial motifs or structural patterns. Some of these motifs are purely structural; others correspond to functional sites.

In order to characterize functional sites, a descriptor is often used that encapsulates the corresponding conserved geometry and/or chemical environment. A comparison between two sites or between a site and a structure is carried out using these descriptors. A functional relationship is then inferred if the sites are significantly similar.

An algorithm for comparing the descriptors is required that identifies geometrical/chemical similarities. Generally, this involves finding equivalent regions in the functional sites (alignment) and scoring the similarities. Finally, a measure of statistical significance is needed in order to identify relevant hits.

There are different possibilities to compare functional sites. The query structure can be either a functional site or a complete protein. This query is compared to each entry in a database, consisting of functional sites or alternatively of complete structures. Databases of functional sites can be defined either manually (more reliable, slow, not reproducible) or with automated methods (faster, reproducible, not restricted to functional sites, problematic data quality).

Different types of descriptors have been proposed. Some are purely geometric, including the atom coordinates, or pseudo-atom coordinates representing the side-chain position, or a surface representation. Others include a description of the chemical environment, either indirectly by considering the amino acid types or, more directly, by considering the chemical functional groups. There are many possible ways to classify these different descriptors. For convenience we define two types; descriptors based on atom coordinates, and descriptors based on chemical environment and surface. Other classifications would be equally valid.

Different applications require different types of descriptors. For example, if side-chain atoms are considered for each residue, the comparison is more specific for a given functional site than if only the coordinates of the C^{α} atoms are used. On the other hand, the former is less robust regarding structural flexibility, vibration, disorder or model quality. When searching for functional similarity between homologous proteins, sequence similarity restricted to the functional site should be taken into account. If the identification of functional similarity on nonhomologous proteins is the goal, then the method should be independent of sequence similarity and the match should rely on the equivalences of chemical functional groups.

When the descriptors are based on atom coordinates, a simple measure like RMSD after optimal rigid-body superposition of the local structures can provide a similarity score for sites with the same size (same number of atoms). Nevertheless, a threshold for the identification of the significant hits is still required. In addition, if the results correspond to matches of different sizes, a measure of statistical significance (p-value or E-value) is necessary in order to compare the results. This is the case when comparison results include partial matches to a descriptor or when they include matches to different descriptors.

3.2 Descriptors based on Atom Coordinates

The most obvious way to represent a functional site is by the coordinates of atoms that constitute the residues making up the functional site. The descriptors vary in terms of whether they use atoms or pseudo-atoms and with respect to the type of chemical information considered. In addition, different search algorithms have been proposed for this type of descriptors.

3.2.1 **ASSAM**

The Willett group was among the early pioneers in the field when they proposed ASSAM in 1994 [8]. The descriptor consists of two or three pseudo-atoms per residue, representing both ends and the midpoint of the side-chain. The query descriptor is searched against a structural database using a standard subgraph-isomorphism algorithm [123]. In the implementation of ASSAM, the pseudo-atoms correspond to nodes in the graph and the distances between the coordinates of these pseudo-atoms correspond to the edges. The method has been successfully applied in the identification of enzymes that include the classical catalytic triad. The triad consist of a conserved serine, histidine and aspartate in the active site of some hydrolases (peptidases and lipases) in different families (nonhomologous, parallel/convergent evolution) [12,26].

A new version of ASSAM has been proposed that features several improvements [113]. In particular, the amino acids have been grouped into different classes according to their chemical properties and matches are allowed between different amino acids of the same class. Secondary structure information has also been added, as well as solvent accessibility, disulfide bridge information and distances to known binding sites.

3.2.2 SPASM

Kleywegt has proposed a related descriptor for the structural comparison of structural motifs [63, 81]. The descriptor is used in SPASM (Spatial Arrangements of Side-chains and Main-chain) to search for matches of a given structural motif in a protein structure database. Each residue is represented by the coordinates of the C^{α} atom and by the pseudo-atom coordinates corresponding to the side-chain center of mass. The method is flexible with respect to the matching of amino acid types. Several options are available; all amino acid substitutions can be allowed, or only some types of substitutions can be allowed, or the substitutions can be scored with a substitution matrix and selected according to a certain cutoff. Additional restrains can be used regarding conservation of sequence directionality, residue neighboring and conservation of gap size.

To search for a functional site in a protein structure, a recursive depth-first search algorithm is used. The candidate residues that match the correct residue type or the allowed substitutions are identified. Then the possible combinations of residues are generated and the distance between atoms (or pseudo-atoms) are compared to the the distances in the query site. If the distances match (differ less than a given cutoff) then the atoms (and pseudo-atoms) are superimposed and the RMSD value is calculated. Results are reported for matches bellow a given RMSD threshold. SPASM reports the possible matches of a query descriptor in a given structural database. RIGOR is a related program that performs the reverse type of search, comparing a query structure to a database of sites.

3.2.3 PINTS

The methods above require that a putative functional site definition is available for the query protein (ASSAM, SPASM) or that a database of characterized functional sites is available (RIGOR). Russell proposed a method for identifying common functional sites between two proteins, without requiring a previous localization of the functional site [102]. He addresses both challenges: localization and characterization. The putative functional site residues are selected according to the established principle of residue conservation and by selection of interacting residues (small inter-atomic distances). In addition, the comparisons are restricted to nonhydrophobic residues and disulfide bridges are ignored. A depth-first search algorithm is used and inter-atomic distances are compared. Unlike SPASM, the proposed method enforces a match of residue types. C^{α}, C^{β} atoms and a single functional atom (predefined according to the residue type) are used in the distance comparison. The matching residues are superimposed and the RMSD calculated. The method also includes a measure of statistical significance (p-values) for the RMSD values. The p-values are calculated based on RMSD values for random matches of side-chain patterns.

An improved model for the statistical significance of RMSD values for local structure comparison has been proposed more recently [116]. The model takes into account the geometry of a match for a given RMSD value, as well as the frequency of residue types and the dependency of the covalently linked atoms in the different side-chains. The derived cumulative distribution functions were shown to fit the observed distribution of the RMSD values for matches between random structural motifs and a background structural database. This model has been included into the search method PINTS (Patterns in Nonhomologous Tertiary Structures). PINTS compares a query site to a structural database, or a protein structure to a database of functional sites [117]. The current limitations of this method are the restrictions to nonhydrophobic residues and the enforcement of exact matches of residue types. Despite these restrictions, PINTS constitutes a reference tool for local structure comparisons and for functional site characterization.

3.2.4 **SuMo**

The SuMo method has been proposed to compare a given query structure to a structural database in order to find similar substructures [53]. The method relies on matching triples of chemical groups. In the comparison of two structures (A, B), a graph of triplets is defined for each structure (S^{A}, S^{B}), where triplets T_1^{A}, T_2^{A} with two common functional groups form connected vertices in the graph. To compare sites in different proteins, the triples are first matched according to the relative postion, local atomic density and type of the chemical groups, resulting in pairs of matching triplets (T_1^{A}, T_1^{B}). Then a comparison graph is defined (G^{AB}), where vertices correspond to pairs of matching triplets (T_1^{A}, T_1^{B}) and edges connect two consistent pairs of triplets $[(T_1^{A}, T_1^{B}), (T_2^{A}, T_2^{B})]$. For the pair of triplets to be consistent, T_1^{A} must be connected by an edge to T_2^{A} in S^{A} and T_1^{B} must be connected by an edge to T_2^{B} in S^{B}, and the angles between the planes of T_1^{A} and T_2^{A} must be similar to the angle between T_1^{B} and T_2^{B}. The independent subgraphs in G^{AB} correspond to similar substructures. As the structure graphs S for all structures in the PDB can be precomplied, it is possible to perform a fast search for all common substructures between a given query and all entries in the PDB. Given a query structure, or a ligand-binding site, the SuMo server [54] can search for all common substructures in the PDB or in a database of ligand-binding sites. Results are sorted by the number of matching functional groups or by RMSD. SuMo does not provide (yet) an estimate of significance of a match. One should also keep in mind that SuMo is a purely structural method; therefore, a match of two substructures does not necessarily imply that they are functional sites, but one can expect that functional sites with significant structural similarity will be detected. The great advantage of SuMo is that it does not require a predefined definition of sites in the query or in the structure database and that the results are provided within seconds or minutes. SuMo relies on matching chemical functional groups, instead of amino acid residues, and does not depend on fold or sequence similarity; therefore, it is suitable to find similar functional sites in the absence of homology.

3.2.5 TESS and Jess

The Thornton group has proposed a method for deriving consensus structural motif descriptors, called templates [127]. The approach was exemplified in the derivation of a template for the classical Ser–His–Asp catalytic triad found in different hydrolases. The template was derived using the coordinates of a known functional site as seed. In particular, the side-chain atoms of the histidine residue were used as reference, and the relative position of the functional oxygens Asp $O^{\delta2}$ and Ser O^{γ} were considered. Then candidate enzymes (serine proteases, subtilisin-like serine proteases, serine carboxypeptidases and lipases) were selected. Interacting Ser, His and Asp were collected and superimposed to the seed. Matching triples (RMSD < 2.0 Å) were used to derive a new consensus functional template. The outcome is a consensus descriptor for the functional site. The method requires some human intervention for the selection of seed structure, the reference and the functional atoms, and the RMSD cutoff for the match. The derived Ser–His–Asp template describes the conserved catalytic triad in the three different types of nonhomologous hydrolases. An additional template that includes the orientation of all side-chain atoms of aspartate and serine (and not only the functional oxygen atoms) was derived for each type of enzyme. The relative orientation of the functional oxygen atoms are conserved across these different enzyme types, but not the overall conformation of the relevant side-chains. Therefore, it was not possible to obtain a consensus template for all enzyme types using all side-chain atoms.

The authors have subsequently proposed TESS – a faster geometric hashing algorithm for the comparison of a functional site template to a structure [126]. To derive a TESS template, a grid is placed around a set of reference residue atoms and the relative positions of the functional atoms are mapped to the grid. In addition, a TESS hash table is obtained for a structural database. This table includes for each occurrence of the reference atoms in the database, the relative positions of the neighboring atoms. The comparison between the query template and the TESS table just requires matching the relevant grid positions, atom types and residue types. If a hit is identified, the coordinates are extracted, and the RMSD of the superimposition is calculated. The method was applied in the identification of different types of catalytic triads involving histidine (including the Ser–His–Asp). TESS allows some flexibility regarding the matching of atom and residue types. The new improved search method, Jess [11], is very flexible regarding the type of descriptor that can be used.

The group has also been compiling a new set of templates for enzyme catalytic residues in the Catalytic Site Atlas (CSA) [99,122] (see also Section 3.4.3). A CSA family has been defined for each group of homologous proteins with a conserved active site. Two different templates were created for each protein family – one based on C^{α}/C^{β} atoms and another based on the functional atoms (that play an active role in the catalysis). The templates allow for matching of chemically similar residues or atoms. The representative template for a family corresponds to the catalytic site with the lowest mean RMSD from all other members of the family. The authors analyzed the diversity of templates within these families. They found that the templates within the homologous families tend to differ by less than 1 Å RMSD and this effect was independent of sequence similarity. The authors also found that templates based on C^{α}/C^{β} atoms tend to be more discriminating than those based on the functional side-chain atoms in the identification of similar catalytic sites within protein families. This is an indication that C^{α}/C^{β} templates are more strongly conserved over homologous families and more robust with respect to structural flexibility than templates based on functional side-chain atoms. Different results might be have been obtained if the analysis had been performed over nonhomologous proteins.

More recently, the authors have derived additional sets of templates in an automated way [71]. Ligand- and DNA-binding templates have been generated from PDB based on the interactions between amino acid residues and small ligands or nucleic acids. In addition, an alternative template search strategy has been implemented. Instead of comparing a query structure to a template library, three-residue templates are extracted from the query structure and compared to a set of representative structures from PDB. This reverse template strategy provides a much higher coverage for matching functional sites than is possible with CSA templates of ligand/nucleic acid templates. The disadvantage is that not all matches correspond to functional sites. The authors have also proposed a new estimate of significance of a match that takes into account the relative positions of the matching residues in protein sequences. This measure is expected to identify similar functional sites in homologous proteins, but is not applicable to compare functional sites in nonhomologous proteins.

3.3 Descriptors based on Chemical Environment and Surface

So far we have described methods that define functional sites in terms of atom or pseudo-atom geometries. An alternative approach is to view the functional site as an environment with certain chemical and physical properties, and with a certain shape. After all, binding is the basis for the molecular function of proteins, and binding specificity is determined by complementarity in terms of shape and chemistry. If the atom coordinates are not taken into account directly, one can expect the descriptors to be more robust with respect to local structural changes and to structural flexibility, and better suited for describing the general molecular function independently of homology.

3.3.1 FEATURE

FEATURE is a method for characterizing the micro-environments surrounding the functional sites in terms of chemical and physical properties [10, 131, 132]. Typical properties are atom types, hydrophobicity, charge, residue types, chemical groups, secondary structure or solvent accessibility. A 3-D grid is used as reference in each protein structure. Each protein atom is mapped to the grid and the values of the corresponding physicochemical properties are stored for each grid cell. Then the spatial distribution of the properties over a certain site can be calculated by summing the property values of all gird cells within a certain radial distance from the center of the site. In this way, a collection of adjacent grid cells is united to a radial shell. The properties pertaining to certain types of sites can then be compared to background properties. If a certain property has a significantly different radial shell distribution in the given site type than in the background (to be determined by a rank-sum test), then this property is considered to be characteristic of that given type of site. In this way the set of property distributions for different radial shells can be identified that are characteristic of a certain type of functional site. These characteristic property distributions can be used as a descriptor in the identification of similar sites in query proteins. In order to compare a query site to a descriptor, a Bayesian scoring function is used to score the likelihood that the property value for a given radial shell in the query site matches the distribution in the site descriptor. The probability values calculated for each property can be combined to give an overall likelihood that the query site matches the site descriptor. A web site is available (WEBFEATURE) to scan a query protein for the occurrence of severall precompiled functional site descriptors. Particularly interesting is the availability of RNA magnesium-binding site descriptors.

A promising development is the use of established sequence motifs as seeds for the definition of the structural-based FEATURE descriptors [76]. In this way it is possible to generate a large number of FEATURE descriptors that are expected to correspond to functionally relevant sites.

3.3.2 CavBase and SiteEngine

The Klebe group has proposed a descriptor for ligand-binding sites that combines physicochemical interaction properties and the surface geometry of the binding site [106]. The ligand-binding sites are extracted using LIGSITE [49], in combination with Relibase, a structural database of protein–ligand interactions [48] (see Section 3.4.1). The physicochemical properties of the binding site are represented as a set of pseudocenters (pseudo-atoms) defined in terms of the atom coordinates of the different chemical functional groups on the protein side-chains and backbone. Five properties are considered: hydrogen-bond donor, hydrogen-bond acceptor, mixed donor/acceptor, aliphatic and aromatic. The binding site descriptor consists of the set of pseudocenters and a grid surface representation. In this surface representation, each grid point is associated with the physicochemical property of the adjacent pseudocenter.

A graph-based approach is used for the comparison of two sites. The problem is reduced to the detection of a maximum common subgraph of the pseudocenters in the two descriptors and solved by a clique detection algorithm [27, 65]. The spatial arrangement of the pseudocenter is viewed as a graph with nodes corresponding to the pseudocenters and edges corresponding to the distances between them. Given two descriptors represented by two graphs, *A* and *B*, all pairs of nodes from *A* and *B* are collected into a product graph *G*. In this product graph the nodes correspond to pairs of pseudocenters with compatible properties. In the product graph *G*, the edges join nodes that correspond to pairs with similar pseudocenter distances in *A* and *B* (difference less than 2 Å). In *G*, the maximum cliques (completely connected subgraphs) are enumerated. Each clique corresponds to a choice of equivalent pseudocenters from both sites with compatible properties and a similar geometry. The size of the clique (number of nodes) is the number of equivalent pseudocenters in each site. The clique solutions are then ranked according to the surface similarity in both descriptors, as measured by the superposition of the surface representation (surface grid points). For each clique, the superposition minimizing the RMSD between the equivalent pseudocenters is calculated. The resulting transformation is applied to the surface grid points and the overlap of surface patches with compatible properties is calculated. Nonoverlapping surface patches and corresponding pseudocenters are filtered out. A new transformation is calculated and the final overlap of compatible surfaces is used to rank the comparisons solutions between two sites. The process is repeated each time a query site descriptor is compared to each entry in a database of binding sites descriptors (CavBase). The surface overlap score and an additional measure are used to compare the search results of different sites. This additional measure takes into account not only the overlap of compatible surface grid points, but also the RMSD of the final superposition of the surface grid.

This group has recently proposed a more efficient similarity search method [133] combining geometric hashing with clique detection. In particular, the method uses a modified product graph in which the nodes correspond to larger matches of local subsets of pseudocenters. This reduces the graph size and the possibility of false matches. The implementation results in considerable speed up of detection with a small sacrifice in coverage.

SiteEngine is a related method developed by the Wolfson group [109, 110]. They also use pseudocenters corresponding to different physicochemical properties and a surface representation. For the comparison they do not use a clique detection algorithm, instead they rely on an efficient geometric hashing implementation. Figure 3 show a comparison of two similar nucleotide binding sites in proteins with different folds. The group also implemented I2I-SiteEngine – a related method for the comparison of protein–protein interfaces [110]. The implementation is similar to SiteEngine, but it additionally relies on predefined patterns of pseudocenters. These patterns correspond to the different types of noncovalent interactions that are expected to be found in protein–protein interfaces.

3.3.3 **eF-site**

The Nakamura group developed the eF-site database, consisting of surface descriptors for different functional sites. The descriptors consist of representations of surface geometry and electrostatics [60–62]. The group also proposed a graph-theoretic based approach for site comparison. Each site is represented as a graph in which each node corresponds to a vertex in the triangular mesh of the molecular surface representation. The product graph corresponds to pairs of nodes from the two sites with similar values for surface curvatures and electrostatic potential. Edges in the product graph represent vertices at similar distances in both functional sites (differing by less than 1.5 Å). A clique detection algorithm [27] is applied to find the similar surface regions.

3.3.4 **pvSOAR**

The pvSOAR (pocket and void surfaces of amino acid residues) method compares protein pockets or cavities regarding sequence, spatial arrangement and orientation [18]. Pockets and cavities are detected by CASTp [75] (see Section 2.3). For each pocket residue, the geometric center of the atoms that contribute to the walls of the pocket is selected. The statistical significance of a match of two pockets or cavities is assessed by aligning the patterns of amino acid pocket residues. The sequence similarity score is

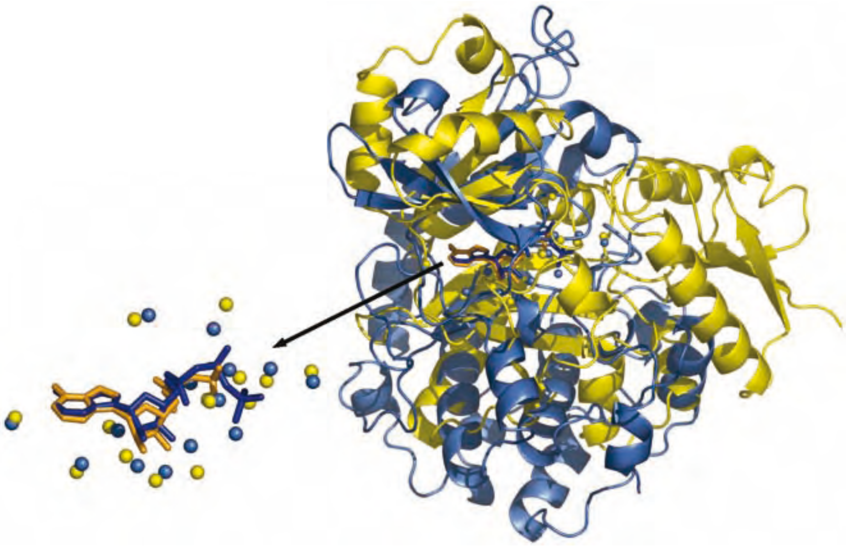


Figure 3 Finding similar small-ligand-binding sites with SiteEngine. The cAMP-dependent protein kinase (blue, PDB code 1atp [139]), and the D-Ala–D-Ala ligase (yellow, PDB code 1iow [39]) have different folds. The kinase binds to ATP (blue) and the ligase binds to ADP (orange). The bound ligands have similar conformations. SiteEngine identified 14 equivalent pseudocenters at the ligand-binding sites. The ligands and pseudocenters are shown enlarged on the left, with the kinase pseudocenters in blue and ligase pseudocenters in yellow. The two structures are shown superimposed according to the transformation from the SiteEngine solution. Graphics produced with PyMOL.

used to compute an *E*-value using a background distribution of similarity scores of randomly shuffled pocket sequences. Two additional measures of significance of match are computed based on the geometry of the matching pockets/cavities. One measure is based on the calculation of the atom coordinate RMSD (cRMSD) after optimal superposition of the atoms in the pocket according to the alignment of the sequence patterns. The second measure is based on the orientation RMSD (oRMSD), which is less sensitive to outliers. The oRMSD is calculated after optimal superposition of coordinate unit vectors defined along the direction of the pocket geometric center. To evaluate the significance of the structural match, *p*-values are computed for both cRMSD and oRMSD, using matches of randomly generated pockets as a background distribution. The amino acid type and position in the sequence is used to compute the sequence-based *E*-value and to define the equivalent residues in the superposition. As the method depends on sequence similarity of the pocket residues, it is expected to perform better in identifying similar functional sites between homologous proteins than between functional sites in nonhomologous proteins. A pvSOAR web server is available [19], where users can compare a given structure to the database of pockets and cavities extracted from the PDB structures. Results can be sorted by sequence based E-value, and by cRMSD or oRMSD p-values, and the aligned pockets are visualized interactively with Jmol (www.jmol.org).

3.3.5 Enzyme Classifier

The development of an enzyme classifier based on self-organizing maps (SOMs) trained with surface cleft properties [114] is an application of statistical learning methods to function prediction based on structural information. Surface clefts were detected with a grid-based method and solvent-accessible surfaces were computed for each cleft. Each surface point was assigned to a chemical type according to the nature and direction of the closest chemical group. Five types were considered: aliphatic, hydrogen-bond donor, hydrogen-bond acceptor, aromatic face and aromatic edge. A set of surface cleft points was used for the generation of topological correlation vectors. Each vector consists of 150 elements; each element corresponds to a certain type of interaction between two chemical types (15 possible interactions), within a certain distance interval (10 equal distance bins of 1.5 Å). Each vector element stores the sum of occurrences of pairs of surface points of the corresponding chemical types and within the corresponding distance. A SOM was then trained using the topological correlation vector and applied in the classification of different types of metalloproteinases.

3.3.6 3D Shape Descriptors

An efficient approach based on spherical harmonics has been proposed for the description of ligand and binding site shape, and for their comparison. The method was applied to virtual drug screening [28] (see Chapter 18). More recently, a similar approach has been proposed for the comparison of ligand-binding sites [82]. The surface is treated as a linear combination of real spherical harmonics. The corresponding expansion coefficients can be used as a description of the shape of the surface cleft and the Euclidean distance metric is used as a measure of similarity between the shapes of the sites. The computation of the expansion coefficients is very fast. Sets of sites can be clustered according to shape similarity using this metric. The approach is quite promising, but there are some limitations. In particular, performance is poor for flat or solvent-exposed binding sites and local matches, where the similarity is restricted to a subregion, cannot be detected. Structural flexibility is not taken into account directly, but it is possible to generate descriptors and perform comparisons for all alternative site conformations given the efficiency of the implementation. Currently, the method is restricted to shape comparison and does not take into account physicochemical properties, but the authors proposed the future inclusion of electrostatic potentials in the descriptor.

3.4 Databases of Functional Sites

Data tend to accumulate dramatically during the process of discovery. Large data volumes can be reached as a result and then it becomes difficult to capture the underling principles from the detailed analysis of single cases. At this point annotated databases and classification methods become valuable, allowing the extraction of the relevant information in a systematic and organized way.

Databases for classification of protein structures [4, 93] fulfilled that important role for the structural data available in the PDB [16]. They provide structural and evolutionary classifications of the known protein structures. However, structures can also be classified based on other principles. A functional classification of protein structures is under way by linking the structural models to functional terms [30, 125]. As mentioned in Section 1.4, molecular function can usually be mapped to a set of localized functional sites. An additional classification database is desirable in this respect, which groups the different types of functional sites according to the underlying molecular function. This will provide a basis for the better understanding of the structure–function relationships and for the development of prediction methods.

Most of the methods for functional characterization described here provide a match between a query protein and a characterized protein. The comparison can be performed between a query structure or site descriptor and a structural database or a database of functional sites. For performance and methodological reasons, it is usually necessary to use a precompiled database of site descriptors, particularly if the query is a complete protein structure. This creates a need for a database of classified functional sites, from which a set of functional descriptors can be compiled. In addition, a database of functional sites can be used to extract the characteristic properties of different functional sites in order to develop new strategies and methods for function prediction. It also affords training, testing and comparing of different methods.

3.4.1 **Relibase**

Relibase is a database of binding sites, ligands and protein–ligand interactions [48]. Structural motifs can be defined and searched for, which is especially interesting in the search for particular protein–ligand interactions motifs. CavBase (see Section 3.3.2) has been integrated into Relibase. The database has been specially targeted to drug design, but it also constitutes a useful resource for general-purpose structural analysis of ligand-binding sites.

3.4.2 **MSDsite**

The Macromolecular Structure Database Group (MSD) at the European Bioinformatics Institute (EBI) provides the community with different types information services for macromolecular structures. One such database is MSDsite [44], an alternative database to Relibase, for the analysis of ligands and ligand-binding sites. Each ligand-binding site in the PDB is extracted and extensively characterized regarding chemistry of the ligand, interacting atoms and residues, and types of interactions. This information is provided for any given PDB structure and the results can be displayed within a molecular viewer. All PDB entries with a given ligand can be easily retrieved. More elaborate queries are also possible, where a given ligand-binding environment is specified and all matching PDB entries retrieved. The environment can be defined by the interacting residue or atom and by the type of interaction (bind length and bond type). In addition to MSDsite, the MSD group provides an extensive set of resources for protein structure analysis [45]. For example, MSDmotif provides extensive search capabilities for structural motifs given sequence patterns, secondary structure patterns or by ϕ, ψ angles and MSDChem allows to search for ligands in the PDB according to given chemical properties.

3.4.3 **CSA**

Enzyme catalytic sites are a particularly relevant type of functional sites. The CSA provides a set of manually curated enzyme catalytic sites, and a set of computed annotations inferred by sequence homology and alignment [99]. CSA includes Jess descriptors (see Section 3.2.5) for some of the homologous families.

3.4.4 **SURFACE**

The SURFACE database (Surface Residues and Functions Annotated Compared and Evaluated) [40] lists the surface clefts on each structure from a nonredundant PDB subset. The clefts are determined with SURFNET [74] (see Section 2.3). Interactions between residues and any bound ligands are identified. The proteins are also annotated with GO terms [29] and with matches to PROSITE [38] sequence motifs. In addition, the authors performed an all-against-all structure comparison of the cleft residues from the different representative structures. Matches are identified according to the residue similarity given by a substitution matrix, as well as according to the Z-score of the RMSD of structural superposition of the clefts. The C^{α} atoms and the average coordinate of the side-chain atoms was used in the superposition. Each PDB structure is associated with the closest representative in the nonredundant set. Users can query the database giving a PDB code, and the information on surface clefts, cleft compassion results, ligand interactions, GO and PROSITE annotation is provided for the representative structure.

3.4.5 Databases of Structural Motifs

Ligand-binding sites can be extracted automatically from the PDB if the ligand is present in the protein structure, as in Relibase or MSDsite, and active sites have been manually compiled in CSA. However, these databases will not include the sites which do not include the ligand in the structural model or which are not yet annotated. Additional databases and approaches are needed to cover these functional sites. A possible strategy is to automatically extract conserved structural motifs from the PDB [24, 47, 53, 87, 128]. The resulting structural motif database is then used to search for similarities to a given query protein structure. It is expected that some (but not all) resulting matches will correspond to functional sites that are not yet included in Relibase, MSDsite or CSA, therefore extending the coverage of functional sites. The disadvantage is that many matches will correspond to purely conserved substructures, without a direct relationship to function. Another possible strategy is the reverse template search [71] described in Section 3.2.5. In this case, structural motifs are extracted directly from the query, therefore avoiding having to define a database of functional/structural sites.

3.4.6 Protein–protein Binding Sites

Apart from ligand-binding sites and the active sites of enzymes, protein–protein binding sites constitute the other major type of functional sites. SCOPPI [135] is a database for protein–protein binding sites and interfaces. In this context, interfaces correspond to the interacting surface regions found on each protein in the complex. Proteins are modular entities composed of domain units. Therefore, the interfaces are characterized at the domain level. Structures are obtained from PDB [16]. The structure of the biologically relevant protein complex is derived from PQS [50] (Section 2.6.2) and domain definitions are obtained from SCOP [4]. SCOPPI classifies the different binding sites within SCOP families into different types (face types). PIBASE and 3did are related protein–protein interaction databases [33, 119].

4 Integration Efforts

As described above, different tools and databases are currently available for locating and characterizing functional sites. Thus, it is natural to ask at this point for the best strategy/protocol to apply given the structure of an uncharacterized protein. Recently, two groups have addressed this question and proposed pipelines for protein function prediction: ProKnow [91] and ProFunc [72].

ProKnow [91] is a pipeline for the annotation of query protein structures with GO [9] functional terms. At the center of the method is the ProKnow knowledgebase which associates protein features with GO terms. Different types of features are extracted from a given query structure: homologous sequences, homologous structures, sequence motifs, interacting proteins and structural motifs (using RIGOR [63], see Section 3.2.2). Then weights for different functional terms are computed using the knowledgebase, by mapping the identified features in the query to functional likelihoods.

The ProFunc pipeline [72] performs three types of analysis for a given query structure: sequence scans, structural features and template searches. In the first type of analysis different sequence-based methods are applied to characterize the query protein. Homologous proteins in UniProt [5] and the PDB are collected based on sequence similarity using BLAST [2]. The resulting alignments are combined to generate a multiple sequence alignment which is used to compute residue conservation scores [124]. The target sequence is also scanned for the occurrence of sequence motifs with InterProScan [138] and for matches in SCOP [4] structural superfamilies with SUPERFAMILY [80]. Gene neighborhood analysis is also performed in order to identify genes functionally related to the target (see Chapter 32). Different structural features are analyzed in parallel. Secondary structure matching (SSM) [67] is used to identify similar folds in the PDB. Surface pockets are located using SURFNET [74] (see Section 2.3). The results from SURFNET together with the residue conservation scores are mapped onto the protein structure for visualization and identification of putative functional sites. A nest search is also performed on the query structure. Nests correspond to a certain type of structural motif associated with anion-binding sites and found often at functional sites [129]. In order to characterize the possible functional sites, Jess [11] is applied to detect matches to different types of templates [71], as described previously in Section 3.2.5. The sets of templates correspond to the manually derived catalytic site descriptors from CSA [99], and to the automatically generated sets of templates for ligand- and nucleic acid-binding sites. A reverse template search is also performed, where the templates are automatically extracted from the target structure and compare to a representative set of PDB structures. Most of the ProFunc computations run in parallel, so that results from the different methods are made available as soon as they are finished.

5 Resources for Structural Characterization

5.1 Available Tools and Databases

Table 1 lists some of the available methods for predicting protein function based on structural information. The methods are freely available, either as software for download or as web services.

Table 1 Tools and databases for structure-based function prediction

Type	Name	Web site	Availability[a]
Functional site localization	SURFNET [74]	www.biochem.ucl.ac.uk/~roman/surfnet/surfnet.html	S
	CASTp [20]	cast.engr.uic.edu/cast	W
	ConSurf [43]	consurf.tau.ac.il	W
	SARIG [3]	www.weizmann.ac.il/SARIG	W
	PPI-Pred [23]	www.bioinformatics.leeds.ac.uk/ppi_pred	W
	PQS [50]	pqs.ebi.ac.uk	W
	NOXclass [140]	noxclass.bioinf.mpi-inf.mpg.de	W
Molecular function characterization	SPASM [81]	portray.bmc.uu.se/cgi-bin/spasm/scripts/spasm.pl	S, W
	PINTS [116]	www.russell.embl.de/pints	W
	SuMo [54]	sumo-pbil.ibcp.fr	W
	FEATURE [10]	feature.stanford.edu/index.html	S
	WEBFEATURE [77]	feature.stanford.edu/webfeature	W
	SiteEngine [110]	bioinfo3d.cs.tau.ac.il/SiteEngine	S, W
	pvSOAR [19]	pvsoar.bioengr.uic.edu	W
	I2I-SiteEngine [110]	bioinfo3d.cs.tau.ac.il/I2I-SiteEngine	S, W
	MSDsite [44]	www.ebi.ac.uk/msd-srv/msdsite	W
	CSA [99]	www.ebi.ac.uk/thornton-srv/databases/CSA	W
	SURFACE [40]	cbm.bio.uniroma2.it/surface	W
	SCOPPI [135]	www.scoppi.org	W
	PIBASE [33]	alto.compbio.ucsf.edu/pibase	W
	3did [119]	3did.embl.de	W
Integrated	ProFunc [72]	www.ebi.ac.uk/thornton-srv/databases/ProFunc	W
	ProKnow [91]	nihserver.mbi.ucla.edu/ProKnow	W

[a] S = Software, W = Web site.

5.2 Characterizing a Protein

In order to demonstrate how these different methods can be used for functional characterization, we applied them to the structure of protein MJ0882. Protein MJ0882 from *Methanococcus jannaschii* has been functionally characterized based on the analysis of the structural model (PDB code 1dus) [52]. Backbone structure comparison revealed that 1dus is structurally similar to several *S*-adenosylmethionine (AdoMet)-dependent methyltransferases, although there was no significant sequence similarity. Manual comparison of the cofactor AdoMet-binding site in methyltransferases and the equivalent region in 1dus revealed considerable similarities. In particular, 1dus displayed the conserved four motifs (motifs I–IV), characteristic of the AdoMet-binding sites. The binding of MJ0882 to AdoMet was then experimentally confirmed, indicating that MJ0882 is an AdoMet-dependent methyltransferase.

First, we applied several methods for functional localization to the 1dus structure. In particular, CASTp identified the AdoMet-binding site as the largest surface cleft. Consurf and SARIG also identified the the cofactor-binding pocket residues. All these web resources for functional site localization provided results within seconds.

Then 1dus was compared to a database of ligand-binding sites with PINTS and the results were also obtained within seconds. The three top ranking hits (lowest *E*-value) are all AdoMet-dependent methyltransferases with significant *E*-values < 0.01 (see Figure 4 for the result with lowest *E*-value). The matching residues correspond to the conserved motifs in the AdoMet-binding site. The fourth-best hit is not a methyltransferase, but its *E*-value (0.02) does not indicate high significance. SuMo rovided results within a few minutes, but does not provide a measure of statistical significance for the hits; therefore, results were not as easy to interpret as with PINTS. When the results are ranked by the number of residues or by volume of the match, among the top three ranking hits there is a match to an AdoMet-dependent methyltransferase. In this solution the AdoMet-binding motifs II and III are aligned. The other hits seem to be false positives. The SiteEngine web service allows the comparison of a structure to a ligand-binding site. 1dus was compared to the AdoMet-binding site of the TaqI DAN methyltransferase, PDB code 2adm chain A [105]. SiteEngine correctly aligned the four motifs in the cofactor-binding site. The entry in the CSA database for 1dus listed residues in motifs I, II and IV, based on homology to the functionals site of another AdoMet-dependent methyltransferase. Within a few hours ProFunc provided a large array of results for the functional characterization of 1dus. Matches to methyltransferases were reported in InterPro [83] and Superfamily [80] using sequence-based methods and by SSM (backbone structure comparison). The nest search method identified the AdoMet-binding motif I, and the AdoMet-

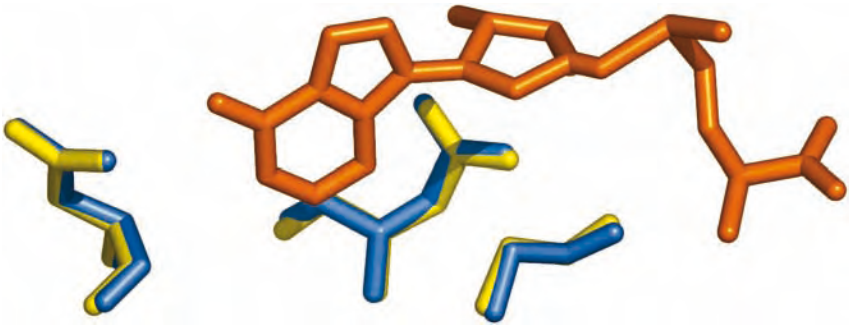


Figure 4 Comparsion of protein MJ0882 (PDB 1dus) with a database of ligand-binding sites with PINTS gives isoflavone *O*-methyltransferase (PDB code 1fpx) as the best result (E-value 3.1×10^{-5}). The side-chains of the three matching residues in 1dus (blue) and 1fpx (yellow) are shown superimposed. The three residues correspond to the conserved AdoMet-binding motifs I, II and III. The cofactor AdoMet from 1fpx is represented in orange.

binding pocket was identified by the residue conservation analysis and by SURFNET.

To summarize, the methods for functional site localization correctly identified the cofactor-binding site. In addition, both PINTS and (less clearly) SuMo results indicate that the protein has a functional site typical of AdoMet-dependent methyltransferases. This was further confirmed by the comparison of 1dus to the TaqI methyltransferase with SiteEngine and by the combined results from ProFunc.

6 Current Applications

There are now documented examples for the application of structure-based function prediction methods in generating valuable hypotheses for function assignment. Some of these results have been used to guide further experiments, which eventually confirmed the original predictions.

The application of the ET method to the RGS signaling family constitutes an early documented success of function prediction. The predicted functional site location was later confirmed by targeted mutagenesis and by structure determination of the protein in a complex [112].

Another example is the functional characterization of the *E. coli* BioH protein [103]. The crystal structure of the uncharacterized protein was determined by X-ray crystallography. Backbone structural comparison results revealed similarities to proteins with different enzymatic functions. TESS was used to compare the structure to catalytic site descriptors compiled from

CSA, resulting in a significant hit to the lipase catalytic triad and indicating hydrolase activity. The protein was subjected to different enzymatic assays in order to test for different types of hydrolase activity. The results indicated that BioH is a carboxylesterase.

PINTS has been applied to structural genomics models, for which sequence-based methods failed to identify any homology to known proteins [115]. The authors compared 157 uncharacterized structures to a database of descriptors corresponding to ligand-binding sites and PDB SITE records. For 17 cases with significant overall fold similarity to a characterized protein, PINTS confirmed the backbone structure comparison result with significant structural matches to functional sites. For 12 query structures corresponding to new folds, new functional hypothesis were suggested. The process is fully automated and can be reproduced as more uncharacterized structures become available.

7 Future Perspectives

In this chapter, we have reviewed a rather young and exciting field in bioinformatics, as can be demonstrated by the fact that more than two-thirds of the references have been published within the last 5 years. The methods available are rather diverse and the approaches quite exploratory. One can expect that as function prediction tools based on structure are more routinely applied and more thoroughly tested, a set of mature procedures will start to emerge. The newly proposed pipelines ProFunc and ProKnow constitute a promising start.

We can also expect that standard test sets and evaluation procedures will be introduced, analogously to the developments in the structure prediction field (see Chapters 10–12). Particularly important in this respect is the further development of databases of functional sites. They are essential for method development, training, testing and application. As we have seen, several functional site databases are already available but, in general, they include a subset of functional site types: catalytic residues (CSA), ligand binding sites (Relibase) or protein–protein interfaces (PIBASE, 3did). There is still a real need for a comprehensive and accurate database of functional sites. In principle, such a database could be implemented manually, but it would be desirable to rely on automated approaches as much as possible in order to reduce the time of data processing, increase coverage and guarantee consistency. In this respect some promising strategies have been proposed to extract conserved structural motifs [24, 47, 53, 87, 128]. The difficulty is then is to distinguish between functional sites and nonfunctional/structural motifs.

A particularly important development is the integration of established sequence-based approaches (Chapter 30) with structure-based methods. These methods go one step further than the identification of homology and directly associate functional terms to the query protein [92, 98].

One can also expect that developments in the fields of structural genomics and structure prediction will boost the range of targets for function prediction. In this respect it is particularly relevant to ask how useful are predicted structural models for functional annotation. The perspective some years ago was not very optimistic [130], but the situation is changing [6]. It is noticeable that predicted models are now being used together with other types of information for the functional annotation of different genomes (see Chapter 12).

It is also reasonable to expect significant contributions to drug design and, conversely, that methods developed for drug discovery to be applied in functional characterization. As an example for the interplay between the two fields, we start to see that application of virtual screening methods (Chapter 18) to the identification of potential natural ligands in functional characterization [56]. Better methods for characterization of protein functional sites will have an impact in the development of new drugs, and in the better understanding and prediction of side-effects.

Acknowledgments

We thank Gerard Kleywegt, Ingolf Sommer, Oliver Sander and Andreas Kämper for their helpfull comments and suggestions. F. S. D. is supported by the BioSapiens project. The BioSapiens project is funded by the European Commission within its FP6 Programme, under the thematic area "Life sciences, genomics and biotechnology for health" (LSHG-CT-2003-503265).

References

1 Aloy, P., E. Querol, F. Aviles and M. Sternberg. 2001. Automated structure-based prediction of functional sites in proteins: applications to assessing the validity of inheriting protein function from homology in genome annotation and to protein docking. J. Mol. Biol. **311**: 395–408.

2 Altschul, S., T. Madden, A. Schäffer, J. Zhang, Z. Zhang, W. Miller and D. Lipman. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**: 3389—402.

3 Amitai, G., A. Shemesh, E. Sitbon, M. Shklar, D. Netanely, I. Venger and S. Pietrokovski. 2004. Network analysis of protein structures identifies functional residues. J. Mol. Biol. **344**: 1135–46.

4 Andreeva, A., D. Howorth, S. E. Brenner, T. J. P. Hubbard, C. Chothia and A. G. Murzin. 2004. SCOP database in 2004: refinements integrate structure and sequence family data. Nucleic Acids Res. **32**: D226–9.

5 APWEILER, R., A. BAIROCH, C. H. WU, ET AL., 2004. UniProt: the Universal Protein knowledgebase. Nucleic Acids Res. **32**: D115–9.

6 ARAKAKI, A. K., Y. ZHANG AND J. SKOLNICK. 2004. Large-scale assessment of the utility of low-resolution protein structures for biochemical function assignment. Bioinformatics **20**: 1087–96.

7 ARMON, A., D. GRAUR AND N. BEN-TAL. 2001. ConSurf: an algorithmic tool for the identification of functional regions in proteins by surface mapping of phylogenetic information. J. Mol. Biol. **307**: 447–63.

8 ARTYMIUK, P., A. POIRRETTE, H. GRINDLEY, D. RICE AND P. WILLETT. 1994. A graph-theoretic approach to the identification of three-dimensional patterns of amino acid side-chains in protein structures. J. Mol. Biol. **243**: 327–44.

9 ASHBURNER, M., C. BALL, J. BLAKE, ET AL. 2000. Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. **25**: 25–9.

10 BAGLEY, S. AND R. ALTMAN. 1995. Characterizing the microenvironment surrounding protein sites. Protein Sci. **4**: 622–35.

11 BARKER, J. A. AND J. M. THORNTON. 2003. An algorithm for constraint-based structural template matching: application to 3D templates with statistical analysis. Bioinformatics **19**: 1644–9.

12 BARTH, A., M. WAHAB, W. BRANDT AND K. FROST. 1993. Classification of serine proteases derived from steric comparisons of their active sites. Drug Des. Discov. **10**: 297–317.

13 BARTLETT, G. J., A. E. TODD AND J. M. THORNTON. 2003. Inferring protein function from structure. In: BOURNE, P. E. AND H. WEISSIG (eds.), *Structural Bioinformatics*. Wiley-Liss, New York, NY: 387–407.

14 BARTLETT, G. J., C. T. PORTER, N. BORKAKOTI AND J. M. THORNTON. 2002. Analysis of catalytic residues in enzyme active sites. J. Mol. Biol. **324**: 105–21.

15 BATE, P. AND J. WARWICKER. 2004. Enzyme/non-enzyme discrimination and prediction of enzyme active site location using charge-based methods. J. Mol. Biol. **340**: 263–76.

16 BERMAN, H., J. WESTBROOK, Z. FENG, G. GILLILAND, T. BHAT, H. WEISSIG, I. SHINDYALOV AND P. BOURNE. 2000. The Protein Data Bank. Nucleic Acids Res. **28**: 235–42.

17 BERNSTEIN, H. 2000. Recent changes to RasMol, recombining the variants. Trends Biochem. Sci. **25**: 453–5.

18 BINKOWSKI, T. A., L. ADAMIAN AND J. LIANG. 2003. Inferring functional relationships of proteins from local sequence and spatial surface patterns. J. Mol. Biol. **332**: 505–26.

19 BINKOWSKI, T. A., P. FREEMAN AND J. LIANG. 2004. pvSOAR: detecting similar surface patterns of pocket and void surfaces of amino acid residues on proteins. Nucleic Acids Res. **32**: W555–8.

20 BINKOWSKI, T. A., S. NAGHIBZADEH AND J. LIANG. 2003. CASTp: Computed Atlas of Surface Topography of proteins. Nucleic Acids Res. **31**: 3352–5.

21 BISWAS, M., J. F. O'ROURKE, E. CAMON, ET AL. 2002. Applications of InterPro in protein annotation and genome analysis. Brief Bioinform. **3**: 285–95.

22 BLOUIN, C., Y. BOUCHER AND A. J. ROGER. 2003. Inferring functional constraints and divergence in protein families using 3D mapping of phylogenetic information. Nucleic Acids Res. **31**: 790–7.

23 BRADFORD, J. R. AND D. R. WESTHEAD. 2005. Improved prediction of protein–protein binding sites using a support vector machines approach. Bioinformatics **21**: 1487–94.

24 BRADLEY, P., P. S. KIM AND B. BERGER. 2002. TRILOGY: discovery of sequence–structure patterns across diverse proteins. Proc. Natl Acad. Sci. USA **99**: 8500–5.

25 BRADY, G. AND P. STOUTEN. 2000. Fast prediction and visualization of protein binding pockets with PASS. J. Comput. Aided Mol. Des. **14**: 383–401.

26 BRADY, L., A. BRZOZOWSKI, Z. DEREWENDA, ET AL. 1990. A serine protease triad forms the catalytic centre of a triacylglycerol lipase. Nature **343**: 767–70.

27 BRON, C. AND J. KERBOSCH. 1973. Algorithm 457: finding all cliques of an undirected graph. Commun. ACM **16**: 575–7.

28 CAI, W., X. SHAO AND B. MAIGRET. 2002. Protein–ligand recognition using spherical harmonic molecular surfaces: towards a fast and efficient filter for large virtual throughput screening. J. Mol. Graph. Model. **20**: 313–28.

29 CAMON, E., M. MAGRANE, D. BARRELL, ET AL. 2003. The Gene Ontology Annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, and InterPro. Genome Res. **13**: 662–72.

30 CAMON, E., M. MAGRANE, D. BARRELL, ET AL. 2004. The Gene Ontology Annotation (GOA) Database: sharing knowledge in Uniprot with Gene Ontology. Nucleic Acids Res. **32**: D262–6.

31 CASARI, G., C. SANDER AND A. VALENCIA. 1995. A method to predict functional residues in proteins. Nat. Struct. Biol. **2**: 171–8.

32 COPLEY, S. D. 2003. Enzymes with extra talents: moonlighting functions and catalytic promiscuity. Curr. Opin. Chem. Biol. **7**: 265–72.

33 DAVIS, F. P. AND A. SALI. 2005. PIBASE: a comprehensive database of structurally defined protein interfaces. Bioinformatics **21**: 1901–7.

34 DENESSIOUK, K., J. LEHTONEN AND M. JOHNSON. 1998. Enzyme–mononucleotide interactions: three different folds share common structural elements for ATP recognition. Protein Sci. **7**: 1768–71.

35 DIETMANN, S., N. FERNANDEZ-FUENTES AND L. HOLM. 2002. Automated detection of remote homology. Curr. Opin. Struct. Biol. **12**: 362–7.

36 DOMINGUES, F. S., J. RAHNENFÜHRER AND T. LENGAUER. 2004. Automated clustering of ensembles of alternative models in protein structure databases. Protein Eng. Des. Sel. **17**: 537–43.

37 ELCOCK, A. 2001. Prediction of functionally important residues based solely on the computed energetics of protein structure. J. Mol. Biol. **312**: 885–96.

38 FALQUET, L., M. PAGNI, P. BUCHER, N. HULO, C. J. A. SIGRIST, K. HOFMANN AND A. BAIROCH. 2002. The PROSITE database, its status in 2002. Nucleic Acids Res. **30**: 235–8.

39 FAN, C., I. PARK, C. WALSH AND J. KNOX. 1997. D-Alanine:D-alanine ligase: phosphonate and phosphinate intermediates with wild type and the Y216F mutant. Biochemistry **36**: 2531–8.

40 FERRÈ, F., G. AUSIELLO, A. ZANZONI AND M. HELMER-CITTERICH. 2004. SURFACE: a database of protein surface regions for functional annotation. Nucleic Acids Res. **32**: D240–4.

41 FLEISCHMANN, W., S. MÖLLER, A. GATEAU AND R. APWEILER. 1999. A novel method for automatic functional annotation of proteins. Bioinformatics **15**: 228–33.

42 GALPERIN, M., D. WALKER AND E. KOONIN. 1998. Analogous enzymes: independent inventions in enzyme evolution. Genome Res. **8**: 779–90.

43 GLASER, F., T. PUPKO, I. PAZ, R. E. BELL, D. BECHOR-SHENTAL, E. MARTZ AND N. BEN-TAL. 2003. ConSurf: identification of functional regions in proteins by surface-mapping of phylogenetic information. Bioinformatics **19**: 163–4.

44 GOLOVIN, A., D. DIMITROPOULOS, T. OLDFIELD, A. RACHEDI AND K. HENRICK. 2005. MSDsite: a database search and retrieval system for the analysis and viewing of bound ligands and active sites. Proteins **58**: 190–9.

45 GOLOVIN, A., T. OLDFIELD, J. TATE, ET AL. 2004. E-MSD: an integrated data resource for bioinformatics. Nucleic Acids Res. **32**: D211–6.

46 GUTTERIDGE, A., G. J. BARTLETT AND J. M. THORNTON. 2003. Using a neural network and spatial clustering to predict the location of active sites in enzymes. J. Mol. Biol. **330**: 719–34.

47 HAMELRYCK, T. 2003. Efficient identification of side-chain patterns using a multidimensional index tree. Proteins **51**: 96–108.

48 Hendlich, M., A. Bergner, J. Günther and G. Klebe. 2003. Relibase: design and development of a database for comprehensive analysis of protein–ligand interactions. J. Mol. Biol. **326**: 607–20.

49 Hendlich, M., F. Rippmann and G. Barnickel. 1997. LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. J. Mol. Graph. Model **15**: 359–63, 389.

50 Henrick, K. and J. Thornton. 1998. PQS: a protein quaternary structure file server. Trends Biochem. Sci. **23**: 358–61.

51 Huang, L., F. Hofer, G. Martin and S. Kim. 1998. Structural basis for the interaction of Ras with RalGDS. Nat. Struct. Biol. **5**: 422–6.

52 Huang, L., L. Hung, M. Odell, H. Yokota, R. Kim and S.-H. Kim. 2002. Structure-based experimental confirmation of biochemical function to a methyltransferase, MJ0882, from hyperthermophile *Methanococcus jannaschii*. J. Struct. Funct. Genomics **2**: 121–7.

53 Jambon, M., A. Imberty, G. Deléage and C. Geourjon. 2003. A new bioinformatic approach to detect common 3D sites in protein structures. Proteins **52**: 137–45.

54 Jambon, M., O. Andrieu, C. Combet, G. Deléage, F. Delfaud and C. Geourjon. 2005. The SuMo server: 3D search for protein functional sites. Bioinformatics **21**: 3929–30.

55 Jeffery, C. J. 2004. Molecular mechanisms for multitasking: recent crystal structures of moonlighting proteins. Curr. Opin. Struct. Biol. **14**: 663–8.

56 Johnston, J. M., V. L. Arcus, C. J. Morton, M. W. Parker and E. N. Baker. 2003. Crystal structure of a putative methyltransferase from *Mycobacterium tuberculosis*: misannotation of a genome clarified by protein structural analysis. J. Bacteriol. **185**: 4057–65.

57 Jones, S., H. P. Shanahan, H. M. Berman and J. M. Thornton. 2003. Using electrostatic potentials to predict DNA-binding sites on DNA-binding proteins. Nucleic Acids Res. **31**: 7189–98.

58 Jones, S. and J. Thornton. 1997. Analysis of protein–protein interaction sites using surface patches. J. Mol. Biol. **272**: 121–32.

59 Jones, S. and J. Thornton. 1997. Prediction of protein–protein interaction sites using patch analysis. J. Mol. Biol. **272**: 133–43.

60 Kinoshita, K., J. Furui and H. Nakamura. 2002. Identification of protein functions from a molecular surface database, eF-site. J. Struct. Funct. Genomics **2**: 9–22.

61 Kinoshita, K. and H. Nakamura. 2003. Identification of protein biochemical functions by similarity search using the molecular surface database eF-site. Protein Sci. **12**: 1589–95.

62 Kinoshita, K. and H. Nakamura. 2005. Identification of the ligand binding sites on the molecular surface of proteins. Protein Sci. **14**: 711–8.

63 Kleywegt, G. 1999. Recognition of spatial motifs in protein structures. J. Mol. Biol. **285**: 1887–97.

64 Ko, J., L. F. Murga, Y. Wei and M. J. Ondrechen. 2005. Prediction of active sites for protein structures from computed chemical properties. Bioinformatics **21** (Suppl. 1): i258–65.

65 Koch, I., T. Lengauer and E. Wanke. 1996. An algorithm for finding maximal common subtopologies in a set of protein structures. J. Comput. Biol. **3**: 289–306.

66 Koehl, P. 2001. Protein structure similarities. Curr. Opin. Struct. Biol. **11**: 348–53.

67 Krissinel, E. and K. Henrick. 2004. Secondary-structure matching (SSM), a new tool for fast protein structure alignment in three dimensions. Acta Crystallogr. D **60**: 2256–68.

68 Landau, M., I. Mayrose, Y. Rosenberg, F. Glaser, E. Martz, T. Pupko and N. Ben-Tal. 2005. ConSurf 2005: the projection of evolutionary conservation scores of residues on protein structures. Nucleic Acids Res. **33**: W299–302.

69 LANDGRAF, R., I. XENARIOS AND D. EISENBERG. 2001. Three-dimensional cluster analysis identifies interfaces and functional residue clusters in proteins. J. Mol. Biol. **307**: 1487–502.

70 LASKOWSKI, R., N. LUSCOMBE, M. SWINDELLS AND J. THORNTON. 1996. Protein clefts in molecular recognition and function. Protein Sci. **5**: 2438–52.

71 LASKOWSKI, R. A., J. D. WATSON AND J. M. THORNTON. 2005. Protein function prediction using local 3D templates. J. Mol. Biol. **351**: 614–26.

72 LASKOWSKI, R. A., J. D. WATSON AND J. M. THORNTON. 2005. ProFunc: a server for predicting protein function from 3D structure. Nucleic Acids Res. **33**: W89–93.

73 LASKOWSKI, R. A. 2003. Structural quality assurance. In: BOURNE, P. E. AND H. WEISSIG. (eds.), *Structural Bioinformatics*. Wiley-Liss, New York, NY: 273–303.

74 LASKOWSKI, R. 1995. SURFNET: a program for visualizing molecular surfaces, cavities, and intermolecular interactions. J. Mol. Graph. **13**: 323–30, 307-8.

75 LIANG, J., H. EDELSBRUNNER AND C. WOODWARD. 1998. Anatomy of protein pockets and cavities: measurement of binding site geometry and implications for ligand design. Protein Sci. **7**: 1884–97.

76 LIANG, M., D. BRUTLAG AND R. ALTMAN. 2003. Automated construction of structural motifs for predicting functional sites on protein structures. Pac. Symp. Biocomput. **8**: 204–15.

77 LIANG, M. P., D. R. BANATAO, T. E. KLEIN, D. L. BRUTLAG AND R. B. ALTMAN. 2003. WebFEATURE: An interactive web tool for identifying and visualizing functional sites on macromolecular structures. Nucleic Acids Res. **31**: 3324–7.

78 LICHTARGE, O., H. BOURNE AND F. COHEN. 1996. An evolutionary trace method defines binding surfaces common to protein families. J. Mol. Biol. **257**: 342–58.

79 MADABUSHI, S., H. YAO, M. MARSH, D. M. KRISTENSEN, A. PHILIPPI, M. E. SOWA AND O. LICHTARGE. 2002. Structural clusters of evolutionary trace residues are statistically significant and common in proteins. J. Mol. Biol. **316**: 139–54.

80 MADERA, M., C. VOGEL, S. K. KUMMERFELD, C. CHOTHIA AND J. GOUGH. 2004. The SUPERFAMILY database in 2004: additions and improvements. Nucleic Acids Res. **32**: D235–9.

81 MADSEN, D. AND G. J. KLEYWEGT. 2002. Interactive motif and fold recognition in protein structures. J. Appl. Crystallogr. **35**: 137–9.

82 MORRIS, R. J., R. J. NAJMANOVICH, A. KAHRAMAN AND J. M. THORNTON. 2005. Real spherical harmonic expansion coefficients as 3D shape descriptors for protein binding pocket and ligand comparisons. Bioinformatics **21**: 2347–55.

83 MULDER, N. J., R. APWEILER, T. K. ATTWOOD, ET AL. 2005. InterPro, progress and status in 2005. Nucleic Acids Res. **33**: D201–5.

84 NEUVIRTH, H., R. RAZ AND G. SCHREIBER. 2004. ProMate: a structure based prediction program to identify the location of protein–protein binding sites. J. Mol. Biol. **338**: 181–99.

85 NIMROD, G., F. GLASER, D. STEINBERG, N. BEN-TAL AND T. PUPKO. 2005. In silico identification of functional regions in proteins. Bioinformatics **21** (Suppl. 1): i328–37.

86 NOOREN, I. M. A. AND J. M. THORNTON. 2003. Diversity of protein–protein interactions. EMBO J. **22**: 3486–92.

87 OLDFIELD, T. 2002. Data mining the protein data bank: residue interactions. Proteins **49**: 510–28.

88 ONDRECHEN, M., J. CLIFTON AND D. RINGE. 2001. THEMATICS: a simple computational predictor of enzyme function from structure. Proc. Natl Acad. Sci. USA **98**: 12473–8.

89 ORENGO, C., D. JONES AND J. THORNTON. 1994. Protein superfamilies and domain superfolds. Nature **372**: 631–4.

90 Ota, M., K. Kinoshita and K. Nishikawa. 2003. Prediction of catalytic residues in enzymes based on known tertiary structure, stability profile, and sequence conservation. J. Mol. Biol. **327**: 1053–64.

91 Pal, D. and D. Eisenberg. 2005. Inference of protein function from protein structure. Structure (Camb.) **13**: 121–30.

92 Pazos, F. and M. J. E. Sternberg. 2004. Automated prediction of protein function and detection of functional sites from structure. Proc. Natl Acad. Sci. USA **101**: 14754–9.

93 Pearl, F., A. Todd, I. Sillitoe, et al. 2005. The CATH Domain Structure Database and related resources Gene3D and DHS provide comprehensive domain family information for genome analysis. Nucleic Acids Res. **33**: D247–51.

94 Perona, J., M. Rould and T. Steitz. 1993. Structural basis for transfer RNA aminoacylation by Escherichia coli glutaminyl-tRNA synthetase. Biochemistry **32**: 8758–71.

95 Peters, K., J. Fauck and C. Frömmel. 1996. The automatic search for ligand binding sites in proteins of known three-dimensional structure using only geometric criteria. J. Mol. Biol. **256**: 201–13.

96 Petock, J. M., I. Y. Torshin, I. T. Weber and R. W. Harrison. 2003. Analysis of protein structures reveals regions of rare backbone conformation at functional sites. Proteins **53**: 872–9.

97 Petsko, G. A. and D. Ringe. 2003. *Protein Stucture and Function.* Sinauer Associates, London.

98 Ponomarenko, J. V., P. E. Bourne and I. N. Shindyalov. 2005. Assigning new GO annotations to protein data bank sequences by combining structure and sequence homology. Proteins **58**: 855–65.

99 Porter, C. T., G. J. Bartlett and J. M. Thornton. 2004. The Catalytic Site Atlas: a resource of catalytic sites and residues identified in enzymes using structural data. Nucleic Acids Res. **32**: D129–33.

100 Pupko, T., R. E. Bell, I. Mayrose, F. Glaser and N. Ben-Tal. 2002. Rate4Site: an algorithmic tool for the identification of functional regions in proteins by surface mapping of evolutionary determinants within their homologues. Bioinformatics **18** (Suppl 1.): S71–7.

101 Russell, R., P. Sasieni and M. Sternberg. 1998. Supersites within superfolds. Binding site similarity in the absence of homology. J. Mol. Biol. **282**: 903–18.

102 Russell, R. 1998. Detection of protein three-dimensional side-chain patterns: new examples of convergent evolution. J. Mol. Biol. **279**: 1211–27.

103 Sanishvili, R., A. F. Yakunin, R. A. Laskowski, et al. 2003. Integrating structure, bioinformatics, and enzymology to discover function: BioH, a new carboxylesterase from *Escherichia coli*. J. Biol. Chem. **278**: 26039–45.

104 Sayle, R. and E. Milner-White. 1995. RASMOL: biomolecular graphics for all. Trends Biochem. Sci. **20**: 374.

105 Schluckebier, G., M. Kozak, N. Bleimling, E. Weinhold and W. Saenger. 1997. Differential binding of *S*-adenosylmethionine *S*-adenosylhomocysteine and Sinefungin to the adenine-specific DNA methyltransferase M. TaqI. J. Mol. Biol. **265**: 56–67.

106 Schmitt, S., D. Kuhn and G. Klebe. 2002. A new method to detect related function among proteins independent of sequence and fold homology. J. Mol. Biol. **323**: 387–406.

107 Schreiber, G., A. Buckle and A. Fersht. 1994. Stability and function: two constraints in the evolution of barstar and other proteins. Structure **2**: 945–51.

108 Shoichet, B., W. Baase, R. Kuroki and B. Matthews. 1995. A relationship between protein stability and protein function. Proc. Natl Acad. Sci. USA **92**: 452–6.

109 Shulman-Peleg, A., R. Nussinov and H. J. Wolfson. 2004. Recognition of functional sites in protein structures. J. Mol. Biol. **339**: 607–33.

110 Shulman-Peleg, A., R. Nussinov and H. J. Wolfson. 2005. SiteEngines: recognition and comparison of binding

sites and protein–protein interfaces. Nucleic Acids Res. **33**: W337–41.

111 SIERK, M. L. AND G. J. KLEYWEGT. 2004. Déjà vu all over again: finding and analyzing protein structure similarities. Structure (Camb.) **12**: 2103–11.

112 SOWA, M., W. HE, K. SLEP, M. KERCHER, O. LICHTARGE AND T. WENSEL. 2001. Prediction and confirmation of a site critical for effector regulation of RGS domain activity. Nat. Struct. Biol. **8**: 234–7.

113 SPRIGGS, R. V., P. J. ARTYMIUK AND P. WILLETT. 2003. Searching for patterns of amino acids in 3D protein structures. J. Chem. Inf. Comput. Sci. **43**: 412–21.

114 STAHL, M., C. TARONI AND G. SCHNEIDER. 2000. Mapping of protein surface cavities and prediction of enzyme class by a self-organizing neural network. Protein Eng. **13**: 83–8.

115 STARK, A., A. SHKUMATOV AND R. B. RUSSELL. 2004. Finding functional sites in structural genomics proteins. Structure (Camb.) **12**: 1405–12.

116 STARK, A., S. SUNYAEV AND R. B. RUSSELL. 2003. A model for statistical significance of local similarities in structure. J. Mol. Biol. **326**: 1307–16.

117 STARK, A. AND R. B. RUSSELL. 2003. Annotation in three dimensions. PINTS: Patterns in Non-homologous Tertiary Structures. Nucleic Acids Res. **31**: 3341–4.

118 STAWISKI, E. W., L. M. GREGORET AND Y. MANDEL-GUTFREUND. 2003. Annotating nucleic acid-binding function based on protein structure. J. Mol. Biol. **326**: 1065–79.

119 STEIN, A., R. B. RUSSELL AND P. ALOY. 2005. 3did: interacting protein domains of known three-dimensional structure. Nucleic Acids Res. **33**: D413–7.

120 THORNTON, J., A. TODD, D. MILBURN, N. BORKAKOTI AND C. ORENGO. 2000. From structure to function: approaches and limitations. Nat. Struct. Biol. **7** (Suppl.): 991–4.

121 TODD, A., C. ORENGO AND J. THORNTON. 2001. Evolution of function in protein superfamilies, from a structural perspective. J. Mol. Biol. **307**: 1113–43.

122 TORRANCE, J. W., G. J. BARTLETT, C. T. PORTER AND J. M. THORNTON. 2005. Using a library of structural templates to recognise catalytic sites and explore their evolution in homologous families. J. Mol. Biol. **347**: 565–81.

123 ULLMANN, J. 1976. An Algorithm for Subgraph Isomorphism. J. Ass. Comp. Mach. **23**: 31–42.

124 VALDAR, W. AND J. THORNTON. 2001. Conservation helps to identify biologically relevant crystal contacts. J. Mol. Biol. **313**: 399–416.

125 VELANKAR, S., P. MCNEIL, V. MITTARD-RUNTE, A. SUAREZ, D. BARRELL, R. APWEILER AND K. HENRICK. 2005. E-MSD: an integrated data resource for bioinformatics. Nucleic Acids Res. **33**: D262–5.

126 WALLACE, A., N. BORKAKOTI AND J. THORNTON. 1997. TESS: a geometric hashing algorithm for deriving 3D coordinate templates for searching structural databases. Application to enzyme active sites. Protein Sci. **6**: 2308–23.

127 WALLACE, A., R. LASKOWSKI AND J. THORNTON. 1996. Derivation of 3D coordinate templates for searching structural databases: application to Ser–His–Asp catalytic triads in the serine proteinases and lipases. Protein Sci. **5**: 1001–13.

128 WANGIKAR, P. P., A. V. TENDULKAR, S. RAMYA, D. N. MALI AND S. SARAWAGI. 2003. Functional sites in protein families uncovered via an objective and automated graph theoretic approach. J. Mol. Biol. **326**: 955–78.

129 WATSON, J. D. AND E. J. MILNER-WHITE. 2002. A novel main-chain anion-binding site in proteins: the nest. A particular combination of phi,psi values in successive residues gives rise to anion-binding sites that occur commonly and are found often at functionally important regions. J. Mol. Biol. **315**: 171–82.

130 WEI, L., E. HUANG AND R. ALTMAN. 1999. Are predicted structures good enough to preserve functional sites? Structure Fold Des. **7**: 643–50.

131 WEI, L., J. CHANG AND R. ALTMAN. 1998. Statistical analysis of protein structures: Using environmental features for multiple purposes. In: SEARLS, D., S. SALZBERG AND S. KASIF (eds.), *Computational Methods in Molecular Biology* Elsevier, Amsterdam: 207–25.

132 WEI, L. AND R. ALTMAN. 1998. Recognizing protein binding sites using statistical descriptions of their 3D environments. Pac. Symp. Biocomput. **3**: 497–508.

133 WESKAMP, N., D. KUHN, E. HÜLLERMEIER AND G. KLEBE. 2004. Efficient similarity search in protein structure databases by *k*-clique hashing. Bioinformatics **20**: 1522–6.

134 WILSON, C., J. KREYCHMAN AND M. GERSTEIN. 2000. Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores. J. Mol. Biol. **297**: 233–49.

135 WINTER, C., A. HENSCHEL, W. K. KIM AND M. SCHROEDER. 2006. SCOPPI: a structural classification of protein–protein interfaces. Nucleic Acids Res. **34**: D310–4.

136 YAKUNIN, A. F., A. A. YEE, A. SAVCHENKO, A. M. EDWARDS AND C. H. ARROWSMITH. 2004. Structural proteomics: a tool for genome annotation. Curr. Opin. Chem. Biol. **8**: 42–8.

137 YAO, H., D. M. KRISTENSEN, I. MIHALEK, M. E. SOWA, C. SHAW, M. KIMMEL, L. KAVRAKI AND O. LICHTARGE. 2003. An accurate, sensitive, and scalable method to identify functional sites in protein structures. J. Mol. Biol. **326**: 255–61.

138 ZDOBNOV, E. AND R. APWEILER. 2001. InterProScan – an integration platform for the signature-recognition methods in InterPro. Bioinformatics **17**: 847–8.

139 ZHENG, J., E. TRAFNY, D. KNIGHTON, N. XUONG, S. TAYLOR, L. T. EYCK AND J. SOWADSKI. 1993. 2.2 A refined crystal structure of the catalytic subunit of cAMP-dependent protein kinase complexed with MnATP and a peptide inhibitor. Acta Crystallogr. D **49**: 362–5.

140 ZHU, H., F. DOMINGUES, I. SOMMER AND T. LENGAUER. 2006. NOXclass: prediction of protein–protein interaction types. BMC Bioinformatics **7**: 27.

34 Mining Information on Protein Function from Text

Martin Krallinger and Alfonso Valencia

1 Introduction

In general, research discoveries made by means of experiments are commonly published in the form of free text in peer-reviewed articles. These articles, stored in scientific literature repositories, are a fundamental data resource for research in molecular biology and biomedicine, and most current research projects start with an extensive trawl through the literature. The scientific literature is consulted by database curators to extract functional annotations, by biologists to obtain information relevant to planning, set-up and interpretation of experiments, and by pharmaceutical companies to aid in the selection of new drug targets. In addition, most current knowledge about the functional aspects of proteins was either directly (by manual annotation during study of the literature) or indirectly (using electronic annotation, i.e. based on sequence similarity to previously annotated proteins) derived from the biomedical literature. Therefore, most bioinformatics tools using functional annotations stored in databases depend on prior extraction of functional descriptions from the literature.

The rapid growth of entries and abstracts in biomedical literature databases such as PubMed (Figure 1), the rapid discovery and analysis of new genes and proteins, and the development and extension of biological ontologies (Chapter 29) have increased interest in ways of extracting functional descriptions automatically from text. Pointers from protein database entries to their corresponding literature, linking sequence information to textual descriptions, are needed. The increasing number of characterized genes together with the growth and accumulation of published biomedical articles makes it difficult for biological databases to cope with the current flow of data.

Additionally the process of populating and updating annotations with information manually extracted from the literature implies considerable delay between the publication and the entering of the annotations in biological databases. Moreover, the curation process is itself very time consuming and costly, requiring highly qualified domain experts.

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

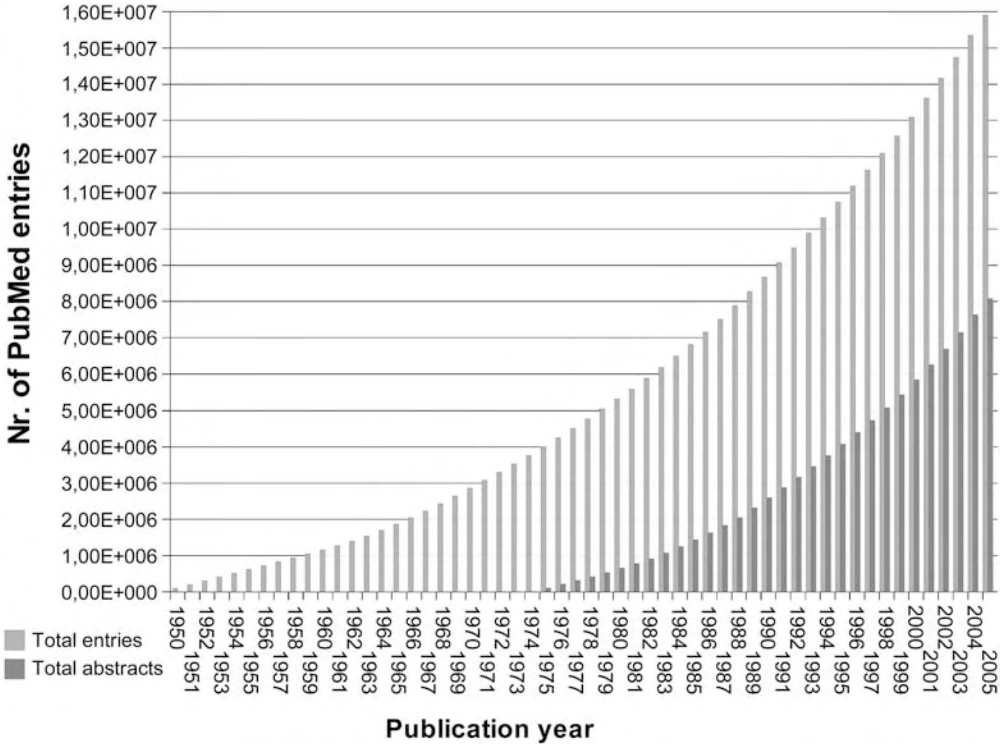


Figure 1 PubMed growth. Growth of the biomedical literature database PubMed in terms of number of total accumulated entries and abstracts for each year retrieved from PubMed using Entrez.

In model organisms such as the mouse genome, a more focused effort is being made to keep existing annotations as complete and up to date as possible (especially for disease-related genes and in some systems such as the immune system), while for other organisms resources are insufficient to keep annotations to the same level, even in cases in which articles providing functional characterizations have been published.

Functional descriptions deposited in the form of database entries therefore represent only a small fraction of existing functional information provided in scientific publications.

Due to the existence of large collections of electronically available articles, computational techniques such as text mining and information extraction (IE) offer the possibility to not only speed-up database curation, helping in the task of converting textual information into structured database entries, but also to provide biologists with better access to functional descriptions extracted from scientific publications. The use of IE techniques during the annotation process can be very useful for maintaining the links between facts stored in databases and the underlying information extracted from the literature, which in turn provide information on the experimental conditions supporting those descriptions, which is usually not incorporated in database annotations.

Text mining and Natural Language Processing (NLP) systems are used for target selection in the case of genomics projects, e.g. FungalWeb [14], and drug discovery [56], complementing traditional bioinformatics strategies to extract those enzymes which are potentially interesting for commercial purposes, for example. In the case of commercial applications in the pharmaceutical industry, text mining has additionally been used in the context of competitive intelligence to monitor competitor information, mine patents, newswire and scientific articles relative to information on drugs, diseases, gene products and chemical compounds, as well as their respective associations. A range of text mining and NLP systems have been recently developed, specifically adapted to the demands and characteristics of biomedical literature, for a review see [56].

This chapter covers basic aspects of data in the form of free text, the most relevant features of information retrieval (IR), text mining and NLP systems designed to provide functional information for proteins and genes. The automatic extraction of different types of text-based functional information is discussed, such as extraction of protein annotations, interactions and subcellular locations from the literature. The most significant resources and methods are introduced briefly. Some of the evaluation metrics and performance assessments currently in use to estimate the performance of existing applications are also introduced.

2 Information Types of Protein Function Descriptions

Different types of biological information can be linked directly or indirectly to protein function. A range of diverse experimental data served to determine protein functions – a considerable number of them are related with the more traditional biochemical and molecular genetics techniques. More recently a full range of technologies in genomics and proteomics (e.g. gene mutation/knockout, coimmunoprecipitation of protein complexes, yeast two-hybrid experiments, gene expression studies with DNA arrays and Chip-on-Chip experiments) have become able to provide additional information on protein function of a substantially different nature. The interpretation and combination of all this information related with protein function can greatly benefit from the use of IE techniques. Information on basically all kinds of experimental results relevant to characterizing protein function is contained in the literature.

Most of the approaches are centered on the analysis of protein sequences and structures (Chapters 30 and 33), and, more recently, on the analysis of gene expression data obtained by microarray experiments (see Chapters 24–27).

Database curators extract annotations supported by experimental evidence from the literature. In the case of Gene Ontology (GO) annotation entries, annotators extract functional information from the literature, providing for each record also the type of experimental information supporting the functional annotation [15]. They thus convert information in the form of free text into entries in structured databases [21]. To populate those database entries it is crucial for annotators to retrieve and extract functional descriptions efficiently from the vast amount of free text. IR, text mining and IE are becoming crucial to save time and effort in manual human curation by providing better access to functional information in biology. For this reason, the GOA database recently included text mining as one of their evidence types for annotation (see also Chapter 29 for more information on biological ontologies).

Humans transmit information by means of natural language expressions to define, communicate and exchange descriptions of functional properties. The way these expressions are formulated often depends on their context, e.g. whether they are designed for annotation records or are embedded in scientific literature. Textual data types often used to describe protein function are basically free-text descriptions, functional keywords and concepts contained in ontologies.

3 Literature Databases in Biomedicine

In order to carry out computational processing of textual data, as is done by NLP, text mining and IR approaches, it is necessary to obtain free text available in a digitally encoded format which can be read by electronic means. One of the first efforts to construct a library of resources available in a machine-readable format was Project Gutenberg. It is still one of the most successful digital library projects, carried out by volunteers who digitize, archive and distribute full-text data, mostly public domain books. Digital libraries are closely associated with the Internet, often being accessed remotely via computer networks. It was, in fact, the introduction of the world wide web and the Internet, where users navigate hyperspace searching for information of interest, which pointed out the importance of efficient IR and web and text mining strategies. Efforts have been made to develop search engines which deal with textual data specifically for academic literature (e.g. books, reports and articles). In this context Google has made an important effort developing

Google Scholar, which allows searching the scholarly literature for relevant research articles, theses or books [13].

Digital libraries have also been constructed for life sciences, especially for the biological and biomedical domain. The most significant of them is the National Library of Medicine (NLM) from the US National Institutes of Health (NIH), consisting of bibliographic databases covering the fields of life sciences, biomedicine as well as other health care and preclinical sciences-related fields.

In life sciences, common knowledge, in contrast to the latest scientific discoveries, is stored in books. The National Center for Biotechnology Information (NCBI), a division of the NLM offers Bookshelf – a collection of biomedical books available online (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi ?db=Books). In addition to books related to medical and clinical subjects, this collection also includes standard biochemistry, molecular and cell biology, developmental biology and immunology books.

The main attention has been paid to processing articles which contain new scientific discoveries (at the time of publication). Citation entries of scientific articles are stored in the PubMed/Medline database developed at the NCBI. This digital library contains more than 15 million citations from over 4800 biomedical journals, most of them (over 12 000 000) articles in English (April 2005). Each entry is characterized by a unique identifier, the PMID. For more than half of them (over 7 000 000), abstracts are available (Figure 1) and often links to the full-text articles are displayed. In the case of old records (before 1975), only relatively few abstracts have been digitized and are provided by PubMed. The PubMed abstracts are currently the most important data source for text mining applications in Biomedicine [53]. According to our estimations over million entries with abstracts contain information relevant to describe important aspects of gene products, such as their function, expression or relationships to diseases. Among the other elements contained in PubMed records are author, journal and title information of the publication. For a (small) number of records gene symbols and molecular sequence databank numbers such as GenBank accession numbers are provided.

In order to characterize articles contained in PubMed, entries have been associated to terms contained in a controlled vocabulary thesaurus with a hierarchical structure – the Medical Subject Headings (MeSH) – allowing searches with those terms at various levels of specificity [71]. There are over 20 thousand descriptors in MeSH and it is used by the NLM to index the entries in PubMed. The use of MeSH terms for text mining purposes is a relevant subject of study in the medical informatics community, as they provide associations of articles to a set of biomedical terms which characterize them.

The PubMed database can be accessed online using a text-based search query system called Entrez and offers additional programming utilities, the

Entrez Programming Utilities (eUtils), to provide a stable interface to this query system. The NLM also leases the content of the PubMed/Medline database on a yearly basis, enabling download of the PubMed records in eXtensible Markup Language (XML) formatted form.

Although some attempts have been made to process full-text biomedical articles [22], the limitations imposed by accessibility and formatting have so far prevented extended processing. Nevertheless, new initiatives related to free and open access of scientific publications are increasing interest in the mining of full-text articles. To provide free access of life science full-text articles, the NIH launched the PubMed Central (PMC) archive in 2000. It contains a few hundred thousand full-text articles as well as book reviews. Efforts are being made by the NIH to digitize older articles, to include new electronic full-text articles and to develop a common format for archiving diverse content provided by different journals in the form of a Journal Archiving and Interchange XML document type definition (DTD). Future developments of PMC will allow searches using the entire body of full-text articles, promoting integration of literature data with other resources such as sequence databases.

4 NLP

NLP methods are being used to handle large collections of electronic articles contained in biomedical literature repositories. NLP is an interdisciplinary research field concerned with the analysis, processing, understanding, modeling and retrieval of natural language. The overall aim of NLP strategies is, despite the complexity of human language, to explore the grammatical, morphological, syntactical and semantic features of well-structured language, in order to process, understand and model natural language by means of computational tools [35].

The statistical analysis of large text collections regarding those features is generally the basic approach used by NLP techniques. Those features are inter-related and often combinations of different features are explored by NLP strategies.

4.1 Grammatical Features

In NLP, grammar refers to the exploration of rules governing a particular language, often referring to those rules which (by convention) are considered to determine the correct formulation of a specific language, such as English. Among the grammatical features studied in NLP is the part of speech (POS) of a word in a given sentence, the role it has in the context of this sentence, whether it is for instance a noun, verb, adjective, adverb or preposition. Programs which automatically label words with their corresponding part of speech in a sentence are called POS taggers. A sample output of a POS tagger corresponds to: *Caspase-3* <Proper noun, sing.> *was* <Verb, past tense> *partially* <Adverb> *activated* <Verb, past part.> *by* <Prep. or subord. Conjunction> *IFN-gamma* <Proper noun, sing.>. [PMID 12700631], where each word in the sentence has been assigned automatically to its corresponding POS label.

POS taggers are usually based on machine learning strategies previously trained using a set of manually POS-tagged sentences. POS information has been shown to be relevant for the identification of gene symbols, which often correspond to nouns or for the extraction of protein interactions from text, where proteins are often associated through a set of interaction-describing verbs.

In order to account for differences in the use of POS for a given word in the case of biomedical texts when compared to generic literature, the Medpost [88] tool has been developed. It is a freely available POS tagger program specifically adapted to biomedical texts. The Medpost authors claim that this system reaches an accuracy of 97% when applied to labeling the corresponding part of speech of words from PubMed abstracts (compared to 86.8% when using a POS tagger trained using generic texts), increasing accuracy over 10%.

4.2 Morphological Features

In the case of morphological features of natural language, word structures are analyzed and rules of how words relate to each other are derived. Those rules include, for instance, plural formation rules (e.g. *gene* and *genes* or *caspase* and *caspases* are singular–plural pairs or the same entity) or verb inflection rules (e.g. *phosphorylate*, *phosphorylates* and *phosphorylating* all refer to the same verb root). Stemmer algorithms are used in this context to normalize word forms to a common stem (word root), providing a way to link different words to the same entity. After applying a stemmer algorithm [78], in the case of the plural formation example, both word forms would map to the same stem, i.e. ‘caspas’.

A common problem associated with stemming algorithms is that they sometimes collapse two words which are semantically different into the same stem (e.g. for gallery and gall).

4.3 Syntactic Features

In order to understand the meaning of a sentence it is necessary to previously identify the relationships between its words, i.e. its syntactic structure. Programs known as shallow parsers analyze such relations at a coarse level, focusing on the identification of phrases (groups of words which function as a syntactic unit) [1]. A sample output using the commercial Connexor shallow parser of the previous example sentence would be:

Caspase-3 <nominal head, noun, single-word noun phrase> *was* <auxiliary verb, indicative past> *partially* <adverbial head, adverb> *activated* <main verb, participle perfect> *by* <preposed marker, preposition> *IFN-* <premodifier, noun, noun phrase begins> *gamma* <nominal head, noun, noun phrase ends>.

Each word is hence labeled according to its corresponding phrase. In particular, phrases where the head is a noun [noun phrases (NP)], "Caspase-3" and "IFN-gamma" in the case of the example, and where the head is a verb [verbal phrases (VP)] are often relevant for NLP applications.

Other useful features include the identification of subject–object relationships from sentences [52]. A sample sentence of this approach, of sentence type NP-VP-NP was: 'Smith and Mitchell (1989) found that [overexpression of <Gene>IMEl</gene>] induced [an <GO>early meiotic event (recombination)</GO> in rich medium], but later meiotic events did not occur (i.e., they detected [no spore formation])'. In this case the subject is represented by the 'IMEl' gene and the object by the Gene Ontology term 'early meiotic event'.

Needless to say, any incorrect identification by the taggers tends to produce a chain of errors during the posterior identification of the syntactic relations.

4.4 Semantic Features

An important issue for NLP strategies is the discovery of associations of words with their corresponding meaning in a given context. To know the semantics (meanings) of a word is a prerequisite to understanding the overall meaning of a sentence. Dictionaries and thesauri provide such associations of words with their corresponding meanings. In the case of the biomedical domain, GO [32] (see Chapter 29) provides a set of concepts which are useful to describe relevant biological aspects of gene products. Also, the collections of gene names and symbols contained in SwissProt and other databases are useful to discover whether in a given sentence a symbol might correspond to a gene or protein. In the case of the following sentence:

Caspase-3 <GENE PRODUCT> was partially *activated* <INTERACTION VERB> by *IFN-gamma* <GENE PRODUCT>.

the associations of words to their corresponding meanings are labeled. Caspase-3 and IFN-gamma are identified as being gene products, and the verb activated in this context refers to a verb which is used to express a certain type of interaction between those two gene products. Gene dictionaries together with the context of occurrence are often indicative of whether a given word or symbol corresponds actually to a gene.

4.5 Contextual Features

When words cooccur significantly often together within a certain textual context (e.g. within the same document, abstract or sentence), they usually display some kind of association. For instance, the cooccurrence of Caspase-3 and IFN-gamma within the same sentence indicates that there is some relationship between them.

To determine whether two documents are similar or whether two proteins are mentioned in texts which are overall similar, the list of words (bag of words) which build up those documents and their associated frequencies (how often they occur within those documents and within the whole set of documents) can be used. The statistical analysis of word frequencies or patterns of word frequencies using large text collections is the base of learning statistical properties of natural language, and characterizing lexical and structural preferences.

As discussed previously, these features are clearly inter-related and the NLP analysis tends to explore the potential of these relations. For example, words which were POS tagged as verbs and which often occur between (connect) gene symbols with functional terms in sentences can indicate a protein description. Similar approaches are commonly used to build patterns of language expression that combine features at different levels. A sample of such language expressions indicating a protein functional description is "protein is involved in".

5 Main NLP Tasks

NLP strategies have been applied to many different tasks, ranging from machine translation (automatic translation of texts from one language into another) to the detection of typing errors in documents. Among the main tasks addressed in NLP as applied to biomedical literature are IR, IE, question answering (QA) and natural language generation (NLG), which will be presented in this section. General techniques and basic terminology used in NLP are described in Table 1.

5.1 IR

To obtain documents related to a certain biological information, such as the interaction between the caspase-3 protein and IFN-gamma, usually in a first step literature databases are consulted to retrieve all the documents which describe this interaction. Thus, all documents which are relevant to this information demand should be retrieved. This is actually the overall aim of

Table 1 General NLP terms and techniques

Corpus: collection of textual data or documents

Indexing: providing terms in textual documents with indices to allow formulation of queries using those terms

Information extraction (IE): analysis of texts to identify entities, relationships and facts in text

Information Retrieval (IR): return relevant documents from a collection of documents according to a predefined information need

Part of speech (POS) tagging: providing each word given a sentence with its corresponding POS label, e.g. whether it is a noun, verb, preposition, article, etc

Question answering (QA): application of computational techniques to generate automatically answers from text collections in response to user queries

Shallow parsing: process of identifying phrasal chunks in sentences like noun phrases, but without providing a deeper grammatical structure

Stemming: process of removing affixes of words transforming them to their corresponding morphological base form or root

Syntax parsing: process of returning the grammatical structure for a given sentence, often in the form of a syntax tree

Tokenization: process of dividing a given text into predefined units, such as words or sentences

Vector space model (VSM): model to calculate the similarity between documents and queries based on the weighted word vectors for terms comprised in the documents

Word sense disambiguation (WSD): process of determining the sense for a word given its context, e.g ATM can refer to the ataxia telangiectasia disorder or the ATM gene, depending on the context used

text IR systems – to retrieve from large documents collections as many relevant documents as possible while retrieving as few as possible nonrelevant ones. To provide relevant documents, those that satisfy a given information demand must be previously identified. In practice, the information demand is formalized or defined in the form of a query. Queries in IR are basically formulations of user demands in the form of words or regular expressions which contain semantic information related to the information need [6]. For instance in our example, a sample query would contain the words "caspase-3" and "IFN-gamma", but it is easy to imagine that users would ideally like to formulate more sophisticated queries such as: "all the documents about capase-3 and IFN-gamma that describes an experimental interaction done *in vivo*".

To know which documents contain the words matching a given query it is necessary to know previously which words build up each document. One of the first steps towards building up an efficient IR system is indexing a collection of documents, locating terms in text [101] and extracting documents in which each term appears. Several strategies exist for locating these terms. The steps common to most of them are:

- Word tokenization (splitting the text into words, e.g. using white spaces for splitting).

- Stop word removal (removing high-frequency words with low information content, e.g. *the*, *a* or *it*).
- Case folding (transforming all upper-case letters to lower-case letters).
- Stemming (transforming words into their basic form by removing affixes).
- Term weighting (assigning a weight to each term according to its relative importance, often based on term frequencies).

These steps help to reduce the vocabulary of terms and the number of query formulations: e.g. the queries *Glycogenin* AND *binding*, as well as *bind* AND *glycogenin*, would retrieve the same documents. The most common strategy used when building the index is based on *inverted file indexing*, in which an index structure that comprises for each term a list of references to documents which contain this term is created [101].

The most widespread information retrieval models are the Boolean model and the vector space model (VSM) – a detailed discussion of different IR models including some probabilistic IR models can be found in Ref. [6]. In the case of the Boolean model, the query is based on the combination of terms using Boolean operators (e.g. AND, OR and NOT). A sample Boolean query would be "caspase-3 AND IFN-gamma", which should retrieve all the documents containing both query words. These query terms are matched against the terms comprised in the inverted file index and a list of documents that satisfy the query expression is returned. The IR system used in PubMed allows such query types [85]. Nevertheless, this type of query usually returns large collections of documents, a considerable number of which are not relevant to the information requested. In addition, relevant documents that do not satisfy exactly the query expression will be absent.

To retrieve documents that are similar to a given list of query terms or that are similar to a given query document, methods based on the VSM are commonly used. In the VSM approach, each document is represented by a vector of terms derived from the document. Each term is weighted according to its relative importance (frequency) within the document and/or within the whole document collection [60]. In addition, the query is represented as a vector of weighted terms. Several different weighting schemes for terms have been developed, but the most widespread are the *tf* × *idf* strategies, which include information on term and document frequencies. The term frequency, $tf_{i,j}$, is the frequency of occurrence of the term j in the document i, whereas the document frequency df_j is the frequency of occurrence of that term in the whole collection of documents. The resulting weight $w_{i,j}$ of term j in document i, given N documents, is obtained by:

$$w_{i,j} = tf_{i,j} \times idf_j \,, \tag{1}$$

where idf_j is the inverse document frequency:

$$idf_j = \log\left(\frac{N}{df_j}\right). \tag{2}$$

As in the case of term weighting, different ways of calculating similarity measures for the term document vectors have been proposed [60]. The cosine measurement is often used to calculate the similarity between the term vector obtained for the query Q and the term vector obtained for each document D in the document collection. Most of them include a normalization factor to avoid the influence of document length. In the cosine measurement, the similarity $\mathrm{sim}(Q, D)$ between the query and a given document is:

$$\mathrm{sim}(Q, D) = \frac{\sum_{j=1}^{V} w_{Q,j} \times w_{i,j}}{\sqrt{\sum_{j=1}^{V} w_{Q,j} \times \sum_{j=1}^{V} w_{i,j}^2}}. \tag{3}$$

The query and the document vectors are represented in a V-dimensional Euclidean space, with V being the number of all terms.

The retrieval strategy behind the *related article search* in PubMed is based on the idea behind the VSM [100], i.e. to provide for a given query document a list of documents ranked according to their similarity to the query document. It allows PubMed document-based queries, returning those records which share relevant terms (words) with the query document on the basis of the frequency of those terms in the whole document (PubMed) collection. The eTBLAST search engine [97] carries out similarity searches using user-provided input text, e.g. paragraphs, whole articles or even proposals, and retrieves similar ones from PubMed.

In practice, not only the calculation of similarities between a single document and each of the documents in the document repository might be of interest. It is also possible, given a set of documents which mention proteins belonging to the same protein family [3] or genes sharing the same cluster derived from a microarray experiment [75], for instance, to automatically extract relevant keywords for this set of documents when compared to the background frequencies of these terms (in the whole document collection).

The performance of IR systems is usually evaluated by means of two relevant feedback parameters, precision and recall. Precision amounts to the number of relevant documents retrieved divided by the total number of documents returned, while recall is the proportion of relevant documents returned by the system divided by the total number of relevant documents. IR performance is often expressed in terms of precision–recall curves. In addition, the f-score, the harmonic mean of precision and recall, is sometimes used to score IR performance [6].

5.2 IE

Although retrieval of relevant articles is one of the first steps to obtain relevant information, these methods do not provide fine-grained identification of specific entities such as genes and proteins, and their respective functional descriptions. For instance, when searching for articles mentioning functional aspects of a specific gene such as *the Drosophila* gene *peanut* [74] using IR systems, many nonrelevant articles are usually retrieved which are related to other topics such as peanut allergy. This is due to the fact that IR techniques do not disambiguate the meaning of words – they are centered merely on the presence or absence (i.e. frequencies) of words and terms. Moreover, most of the genes have a range of synonyms or can be mentioned using the full name or one of their corresponding gene symbols [19]. Therefore, to retrieve all the articles describing functional aspects of genes would imply first searching using all the different combinations of gene names, symbols and typographical variants of these gene symbols, and then selecting those documents which actually are related to the gene of interest. IE methods are useful to avoid this time-consuming process and automatically mine sentences or text segments containing specific entities of interest (e.g. proteins, genes, drugs). For instance, an efficient IE system would identify that the first of the following sentences mentions the peanut gene, while in the second sentence the peanut plant (*Arachis hypogaea*) is mentioned:

(i) *The* Drosophila peanut *gene is required for cytokinesis and encodes a protein similar to yeast putative bud neck filament proteins* [PMID 8181057].
(ii) *In this study, we identified a novel PLD gene in peanut (*Arachis hypogaea*), encoding a putative phospholipase D (PLD, EC 3.1.4.4)* [PMID 16200410].

IE refers to the automatic identification of entities [named entity recognition (NER)] of relationships between those entities, as well as facts and events from unstructured text. IE methods applied to generic or newswire texts have been used to identify company names, persons, locations or events such as terrorist attacks from text. In fact, the identification of entities (NER) in text has been addressed by IE systems for years [39]. In the pharmaceutical industry, IE systems are being used not only to identify textual passages referring to gene names and drugs, but also extract information relative to competitor companies and their current research interests from scientific articles, newswire texts and patents.

In the case of academic IE applications developed for the life science literature, they are mainly related to the identification of biological entities such as genes, proteins, cell lines and chemical compounds, as well as the extraction of protein interactions and protein functional annotations [9].

IE strategies exploit all kinds of features of natural language, including syntactical, lexical and contextual features to identify and characterize relevant biological information. Among the most frequently used NLP techniques in IE are POS tagging, shallow parsing and stemming (Table 1).

Often, certain additional features (e.g. use of capital letters and hyphens) are explored by IE systems designed to identify entities such as gene names [39]. Some IE strategies use rule-based approaches to detect certain word patterns which are often encountered within gene names [34], or language expressions indicating protein interactions [76], phosphorylation [46] or protein functions [25].

Although some strategies generate rules automatically [92], most of them rely on rules generated manually by domain experts [10,52].

Domain-specific knowledge is used by IE systems to account for linguistic properties that are characteristic of the scientific literature. Many IE procedures take advantage of knowledge representations such as ontologies (e.g. GO) for the extraction of functional descriptions or annotations of proteins from texts [80,91].

In the case of the identification of gene names, IE methods can rely on the availability of gene lists provided by biological databases [34] or use machine learning techniques (trained on manually gene-tagged texts) such as support vector machines (SVMs) [67] (see also Chapter 26) or hidden Markov models (HMMs) [50] (see also Chapter 3), identifying also gene names which are not yet included in existing biological databases.

A more detailed discussion of different applications of IE systems is provided in Section 6.

5.3 QA

To obtain information, humans formulate questions using natural language expressions such as: "(i) *What are the molecular functions of Glycogenin ?*" or "(2) *Which organisms express glycogenin?*". The field of QA addresses the automatic generation of answers to specific user queries formulated as natural language expressions to large document collections [31]. Most of the existing QA techniques are limited to general literature or newswire texts and have not yet been tailored to the biology-specific literature. In the best of the cases they are only effective for questions that can be answered with few words derived directly from text passages. The development of QA systems in biomedicine and molecular biology is especially cumbersome because these are poorly formalized, very heterogeneous domains and new scientific terms are constantly created. *Question parsing* is one of the steps used by QA systems, consisting of the analysis of the query sentence with respect to its semantic structure (semantic representation). In the first example question, two elements are significant, i.e. the entity glycogenin (protein) and its molecular function. The next step would involve question analysis of the input question to obtain the relevant content words to perform IR, e.g. the words *glycogenin* and *functions*. In this step, the words or tokens to be used for IR searches are determined. Then, the actual document retrieval using those words is carried out, yielding a (ranked) list of relevant documents. In this case, the actual query could consist of *glycogenin* AND *function*. The retrieved documents are then split into segments or passages to avoid processing of long documents and to operate by using only the relevant segments (document segmentation step). To extract those segments which might contain the desired answer (which contain the query entities), the segments are ranked and each segment is further processed at the sentence level to identify relevant entities. Then, the question and the potential answer segments are compared (word overlap) and ranked according to their word usage similarity to the input question. Words can also be weighted taking into account their background frequency within the whole document collection. Finally, the potential answers are ranked, visualized and presented to the user. In the case of the used example, the following PubMed abstract sentence could be presented: *Glycogenin is a glycosyltransferase that functions as the autocatalytic initiator for the synthesis of glycogen in eukaryotic organisms* [PMID 15849187].

One of the few QA systems in the biomedical domain was developed by Galitsky [30]. This system is based on the use of semantic skeletons (SSK), which consist of matching the semantic representations of a given question to that of the set of potential answers. It thus tries to match the formal representation of a question with the formal representation of an answer. By introducing logical programming strategies it was also able to handle semantic rules.

Many of the existing QA systems for the genomics domain were developed in the context of the Text REtrieval Conference (TREC) Genomics track – a contest to evaluate information retrieval and QA techniques applied to the biology literature. The Genomics track is one of the areas addressed at TREC – a periodically organized workshop where previously posed tasks of IR and QA for different areas (including spam filtering and IR at a terabyte level) are evaluated. At the *ad hoc retrieval task* of the TREC Genomics Track 2005 [37], a collection of sample topics containing a topic title, information need and topic context (formulated as natural language expressions) was given to the participating systems, which had to return the PubMed documents relevant to this information need. For instance, one of the topic titles was: "Find information about base sequences or restriction maps in plasmids that are used as gene vectors".

5.4 NLG

The output of QA systems consists of those sentences in text collections that hold relevant information to the question formulated. There is increasing interest in the construction of natural language texts as output of QA systems, to facilitate to the end user with the interpretation of the information provided. Displaying database contents or the results of gene annotation efforts as automatically generated reports or summaries is also attracting some attention.

NLG is concerned with the development of natural language texts by computer programs based on a semantic input, i.e. providing a computer-internal representation of the information [24]. NLG systems show different degrees of complexity, ranging from *canned text systems*, which return unaltered sentences without any change, to complex *feature-based systems*. NLG has also importance for automatic generation of summaries. Kang and Park developed a system for automatically constructing gene summaries from articles by selecting and ordering those sentences which contain gene related facts [48a]. Therefore they also took into account aspects related to the arrangement of sentences to provide a coherent discourse of the summary. Currently, NLG tools are only efficient in the case of very specific applications. In the case of biology-related texts, the additional difficulty of modeling domain language has to be considered.

Only a limited number of NLG systems have been applied in the biological domain. The Simpathica/XSSYS trace analysis tool combines a natural language query system and a story generation system in a bioinformatics tool devoted to the analysis of biological data such as those resulting from microarray time course experiments [4]. It uses prepositional temporal logic and generates sentences of biologically relevant facts using a set of heuristics.

6 Difficulties when Processing Biological Texts

Computer programming languages are aimed at providing a standardized way of unambiguously expressing instructions to a computer using well-defined syntactic and semantic rules. In contrast, human language is characterized by complex grammatical constructions, and both semantic and syntactic ambiguity. In the case of semantic ambiguity, a given word can have different meanings (polysemy) depending on the context it is used in or even on the background knowledge of the person who is interpreting it. An example would be the previously mentioned case of the word “peanut”, which depending on the context could refer to a *Drosophila* gene, a plant or a plant seed. To automatically identify the correct meaning of words (word sense disambiguation) is especially cumbersome in cases where these words are used in a similar contexts (e.g. protein symbol and the associated disease name). There is also a significant semantic variability in natural language, meaning that two different words (synonyms) or even text fragments (textual entailment) can have the same meaning. The words *apopain* and *Yama* correspond to synonyms of the same human gene, i.e. *CASP-3*. For the identification of gene synonyms most methods rely on the synonym lists provided by biological databases, which are often incomplete. Also, novel names and terms are constantly created and used in the literature which are not included in any of the existing dictionaries.

Human language allows us to express the same meaning in a range of different ways. To infer (entail) that one text fragment has the same meaning as another text fragment is especially cumbersome [23]. This scenario is one of the main challenges for systems which try to associate proteins to functional terms such as GO terms, as a variety of different language expressions may display the same meaning of a single GO term. The following two sentences illustrate this difficulty:

(i) *Fim1 is involved in cell division*.
(ii) *Fim-1 function in cytokinesis*.

In this case both sentences have essentially the same meaning, but only in the second case is the standard GO term *cytokinesis* used.

Another difficulty is syntactic ambiguity, i.e. sentences can be interpreted in different ways depending on how the reader interprets the syntactic connections between words and arranges them into sentences. The sentence "*We analyzed the protein with a functional complementation*" could be interpreted as:

(i) *The protein was studied using the functional complementation technique*, or
(ii) *The protein which displays functional complementation was studied*

To avoid repetitions and to improve writing style, pronouns are commonly used to refer to previously mentioned nouns. For instance the word "*These*" refers to the proteins Bax and Bad in:

(i) Bcl-2 and Bcl-x_L expressions were down-regulated after paclitaxel treatment in FHIT-expressing cells, whereas Bax and Bad expressions were up-regulated.
(ii) These were reversed by siRNA treatment.

To identify those relationships between pronouns and nouns, known as anaphora resolution, can be arduous, especially in long and complex sentences, and when the noun and pronouns are located far from each other within the text. Authors of anaphora resolution systems developed for biomedical literature claim that they reached a precision of 77% and recall of 71% [16].

Another difficulty arises because most words in texts occur extremely infrequently (data sparseness) and have an uneven distribution. Only a small number of words, known as *function words*, like prepositions, appear with high frequency and are very common. Zipf's law describes this characteristic of frequency distribution of words in human language, where a few words appear very often, a middling number corresponds to medium frequency words and the majority are low-frequency words [63, 82]. It is difficult to make predictions of word behavior when relevant words are rare or often not picked up in the data set used to derive the statistical language models, but do appear in data used to test the model. Word frequency distributions are not only affected by the size of the dataset used, but also by the corresponding domain language.

Most of the currently available NLP technologies were developed using generic texts and word frequency distributions differ depending on the domain. Thus, the performance of generic tools is lower when used for biomedical literature.

The majority of life science articles published since the 1950s are in English, but more than half of the corresponding authors have a different native language. The writing style of authors has been shown to depend upon their native language, resulting in variations of word and phrase usage as well as writing style and average sentence lengths. In some cases, this lack of a standard scientific English can reduce the understanding and processing of scientific communications [73].

Mining of biology literature is especially challenging, as it is a rapidly evolving, poorly formalized domain with ambiguous and flexible naming of entities. Often, new gene names are created which are not contained in gene lists of any biological database. The domain is characterized by a great deal of domain-specific terminology (over 12% consists of domain-specific terms) [89], but existing dictionaries are incomplete and standard terminology is rarely used. Some systems tried to circumvent this drawback by developing automatic term recognition systems [72]. NLP tools, which process biomedical literature, have to be adapted taking into account knowledge and features which characterize texts of this domain.

Low-level processing, even tokenization, i.e. dividing texts into units such as words or sentences, is cumbersome in scientific texts due to the heavy use of punctuation marks and special characters. Anyhow, an increasing number of strategies are being developed to solve the important task of extracting functional information from life science texts. Performance of such systems is increasing not only due to adoption of more sophisticated approaches, but also due to the availability of new textual training data which allows the training and comparison of alternative methods. The following section provides an overview of the different applications which have been developed in this field.

7 Strategies of Extracting Functional Information from Text

A variety of strategies have been devised to associate proteins with functional information derived from scientific text elements which can be processed by computational tools.

There are two basic types of text mining approaches supplying functional information through literature analysis. The first comprises direct function extraction methods that identify functional annotations from text, i.e. associations of proteins with functional expressions (keywords, concepts, phrases) using IE, heuristics, statistical and machine learning techniques and rule-based strategies. The second type of approach comprises indirect function extraction methods that discover characteristics of proteins related to the biological context in the whole cell. These indirect methods include the text-based discovery of protein interaction partners and networks, the cellular localization of proteins [90] and the extraction of kinetic parameters [33]. Both methods, as well as the previous tagging of proteins in text, are discussed here.

7.1 NER and Protein Tagging

After retrieval of an initial set of documents that might hold relevant information for proteins, it is crucial to precisely identify those entities within the articles and to link them to entries in biological databases. The second step is called NER. While the first step of name identification is generic, i.e. basically the same in a wide variety of domains, the second step differs between domains. In biology it requires the precise identification of the individual gene (and not the generic name of a family) and the corresponding organism.

NER has been studied by the IE community for years. In the case of entities such as corporate names and locations, NER techniques achieve an f-score of over 0.9 [105]. The recognition of biological entities is far more complex and the results in protein tagging are clearly worse than in the identification of other entities like corporate names.

Difficulties in identifying correctly protein names in texts are of various types. Authors who refer to genes and proteins in articles often do not use the official gene symbols, but express those entities using synonyms or full gene names instead [19, 41]. The disambiguation of gene names that correspond to medical terms or common English words, such as many fly genes, e.g. *18-*

wheeler or *amnesiac*, and the existence of alternative typographical variants of gene names constitute additional hurdles. Even associating genes with the corresponding source organism is cumbersome, as 14% of genes are estimated to display inter-species ambiguity [19]. Finally, the ambiguity between distinct protein names and their protein family names lowers the performance of NER systems in the biology domain [56] [e.g. protein kinase C (PKC) family and individual members such as PKC-α, -βI or ζ]. Also new gene names appear in texts which are still not contained in any biological database – to identify those novel gene symbols is only possible using methods which are able to use inference from the context.

The performance of top-scoring NER systems that tag protein and gene names ranges from f-scores of 0.7 to 0.9 depending on the organism source of the gene names [38], the method used as well as if distinction between identification of proteins, DNA and RNA was made [50]. Nevertheless, this performance was reached using documents which had been previously filtered and were known to be associated to certain model organism genes. Therefore, when randomly selecting a document from the PubMed database and tagging gene and protein names, the overall performance of these systems would be lower.

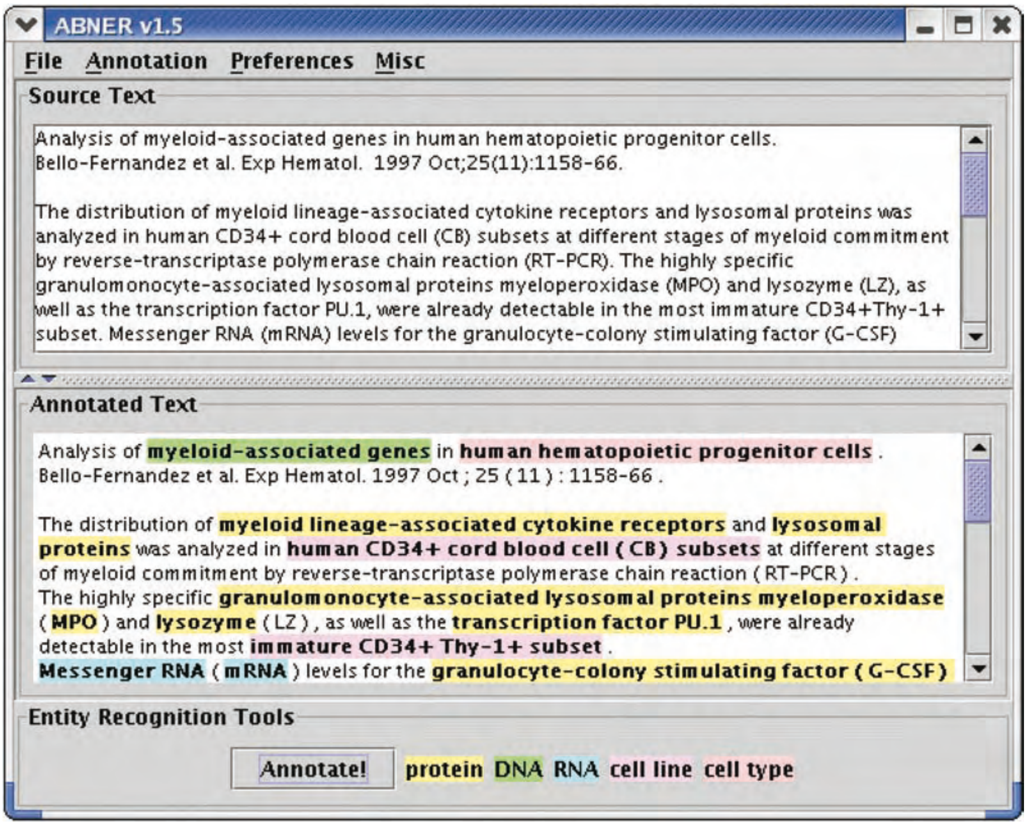
When addressing the NER of proteins, two basic perspectives can be distinguished. The bioinformatics perspective (gene dictionary based) focuses on the exploration of typographical variants of gene names and symbols extracted from biological databases and the use of approximate string matching [57]; whereas the computational linguistics perspective (NLP based) uses linguistic aspects such as POS information [29]. Most of the currently available tools are hybrid systems combining characteristics of both approaches. They integrate methods such as statistical analysis, machine learning techniques, rule-based strategies, morphological features, context and lexical exploration. A sample case of the first perspective is the ProMiner system [34] – a rule-based strategy which also integrates a disambiguation procedure to associate a detected gene to its database entry. To associate genes to database entries, the contextual words in which these gene names occur as well as information about the organism source of the gene symbol is generally used. The disambiguation between mouse genes and human genes is in practice especially difficult, as many homologous genes have the same symbols in both organisms; additionally, often within a given text passages both organisms are mentioned. In the case of machine learning strategies the PowerbioNE system is one of the most effective systems to detect biological entities [107]. It uses morphological, grammatical and syntactic features which are integrated into an hidden Markov model (HMM) entity recognizer. Although a number of NER systems have been constructed for the biology domain, only few of them are available and used in practice. Some of them are available as online taggers such as GAPSCORE [17], a SVM-based system which scores each word of an input text based on statistical models and permits a cutoff setting different degrees of stringency to be defined. Some NER systems can be integrated into local text mining systems. The ABNER open-source tool [86] uses machine learning techniques, i.e. conditional random fields (CRFs), to tag biological entities. It identifies protein DNA, RNA, cell lines and cell types (Figure 2). This system provides an easy-to-use graphical interface and incorporates routines to train new modules on other text collections. Another system which has been used to identify bio-entities is the AbGene system [92] – a program which uses POS information, manually generated rules, and suffix and context information to identify potential gene names. In addition, it applies naive Bayes learning to analyze the whole document context for the likelihood of whether it contains gene names. The drawback of these two systems is that they lack a linking procedure for connecting identified proteins to existing database entries. A useful tool which also carries out the database-linking step and can be run locally is called NLProt [67]. It combines SVMs with dictionary and rule-based filtering to identify protein names in biomedical articles, but lacks identification of other entities such as cell types. A sample output of the NLProt system would be:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<abstract id="">
<protname score="1.023" method="SVM" dbid="ICE3_HUMAN" idreliab="100%"
org="homo sapiens">Caspase-3</protname> was partially activated by <protname
score="1.105" method="SVM" dbid="ING_HUMAN" idreliab="100%" org="homo
sapiens">IFN-gamma</protname>.
</abstract>
```

NLProt also provides a reliability score for the identification and linking of the recognized proteins.

Other useful biological entities are chemical names. useful biological entities are chemical names. The identification of chemical concepts in text was studied by Wilbur and coworkers [99]. They tested various strategies and implemented a statistical method that could reach 97% classification accuracy.

Users often like to search for related articles by means of a query protein sequence rather than protein symbols. MedBlast is a literature-mining tool which allows both search strategies for retrieval of relevant articles, i.e. searching with a query sequence or using gene symbols [93]. It integrates a BLAST search of GenBank and then retrieves the relevant articles for genes with a significant sequence similarity directly from PubMed. It thus integrates both sequence searches with protein name identification.



Analysis of myeloid-associated genes in human hematopoietic progenitor cells.
Bello-Fernandez et al. Exp Hematol. 1997 Oct;25(11):1158-66.

The distribution of myeloid lineage-associated cytokine receptors and lysosomal proteins was analyzed in human CD34+ cord blood cell (CB) subsets at different stages of myeloid commitment by reverse-transcriptase polymerase chain reaction (RT-PCR). The highly specific granulomonocyte-associated lysosomal proteins myeloperoxidase (MPO) and lysozyme (LZ), as well as the transcription factor PU.1, were already detectable in the most immature CD34+Thy-1+ subset. Messenger RNA (mRNA) levels for the granulocyte-colony stimulating factor (G-CSF)

Analysis of **myeloid-associated genes** in **human hematopoietic progenitor cells** .
Bello-Fernandez et al. Exp Hematol. 1997 Oct ; 25 (11) : 1158-66 .

The distribution of **myeloid lineage-associated cytokine receptors** and **lysosomal proteins** was analyzed in **human CD34+ cord blood cell (CB) subsets** at different stages of myeloid commitment by reverse-transcriptase polymerase chain reaction (RT-PCR) .
The highly specific **granulomonocyte-associated lysosomal proteins myeloperoxidase** (**MPO**) and **lysozyme** (LZ) , as well as the **transcription factor PU.1** , were already detectable in the most **immature CD34+ Thy-1+ subset** .
Messenger RNA (**mRNA**) levels for the **granulocyte-colony stimulating factor (G-CSF)**

Figure 2 ABNER. NER system of biological entities such as genes, proteins, DNA and cell lines based on CRFs.

7.2 Associating Proteins with Biological Features from Databases and Ontologies

Database curators extract functional protein annotations manually through a time-consuming and labor-intensive process of scientific literature analysis. Due to the vast amount of literature currently available and the increasing number of genes studied, humans are only able to review a small fraction of the existing publications. Thus, the manual annotation process lags behind the information that accumulates in biomedical articles. Text mining and IE methods have been developed. to avoid this bottleneck and to assist in the identification of functional descriptions of proteins and in automatic annotation extraction.

Procedures extracting functional descriptions of proteins rest upon characteristics of IE and IR methods, and exploit all the different features of natural language, especially grammatical, syntactic and semantic features.

There are two basic types of text mining strategies designed for functional annotation extraction, although many tools combine aspects of both:

- Identification of annotation-relevant sentences: methods which automatically classify text passages or sentences according to whether they contain functional descriptions of proteins.

- Identification of protein-term associations: extraction of relations between proteins and predefined terms, such as concepts from biological ontologies (e.g. GO) or functional keywords from biological databases (e.g. Swiss-Prot keywords).

In the first approach, the context information provided by text passages or sentences in which the proteins are mentioned aids in the appraisal and interpretation of the descriptions. In addition, as descriptions are not limited by a predefined set of functional terms, they account for the variety of language expressions involved in functional descriptions. The main drawback of these strategies is their limitation when inferring function from electronic annotation (i.e. based on sequence similarity) as they do not provide associations between proteins and controlled vocabulary concepts (e.g. GO terms) as well as the implicit difficulty while dealing with certain uninformative words. This means that the extraction of annotation-relevant sentences requires human interpretation.

The system developed by Andrade and coworkers [3] extracts functional information from PubMed sentences by means of statistical analysis of the word frequencies of sentences related to each protein family compared to the background frequencies of those words for the whole set of protein families. It thus detects biologically significant words useful to describe functional aspects of protein families. An inherent difficulty of this approach is the existence of words which are statistically associated to a protein group, but are not useful for describing functional aspects.

Beyond the level of single words, the identification of consecutive word patterns, which are often used in functional descriptions of proteins in abstracts, has been studied [55]. In the presence of protein names, these patterns indicate that a functional description is provided, e.g. "*gene is essential for*". Such approaches are useful in combination with NER systems to classify whether sentences correspond to functional descriptions. The main limitation of such techniques is related to the construction and evaluation of extensive lists of word patterns, which is a time-consuming process and requires domain expert knowledge.

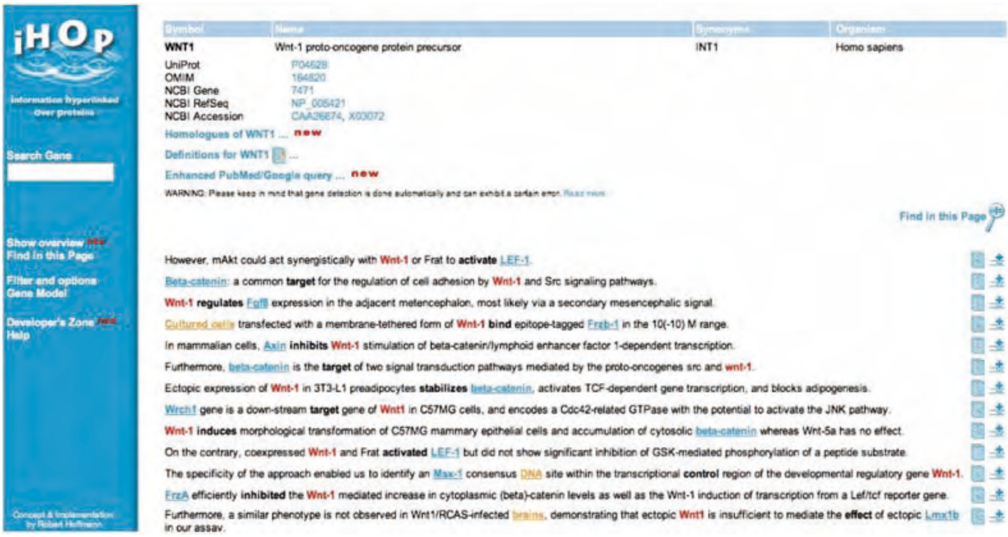
A more sophisticated approach was used by the BioIE system – a rule-based method for identification of informative sentences [25]. For a given query term (e.g. gene names) it returns a list of sentences which match a set of manually defined templates and rules for the extraction of informative sentences. The identified templates are highlighted within those sentences and information of the word distributions is also provided. This system (available as an online server), focuses on the extraction of sentences with information related to predefined categories, i.e. structure, function, diseases and therapeutic compounds, localization and familial relationships. BioIE has been integrated into a web-based annotation tool called METIS. METIS performs for a given query sequence a BLAST search against the Swiss-Prot database. Then, for the retrieved entries, a structured report is generated and informative sentences are extracted from PubMed using BioIE as well as a SVM-based sentence classifier.

The use of a curated set of concepts and functional terms with established value for protein annotation is one of the main advantages of the second strategy. These concepts can be used to transfer function by electronic annotation and, in the case of terms derived from ontologies, even allow inference of other functional terms taking into account the ontology structure. However, the use of a limited set of concepts is a restrictive approach, as the concepts only correspond to a fraction of the possible ways of expressing function in text due to the semantic variability of natural language. Moreover, concepts derived from ontologies or thesauri often do not resemble natural language expressions (which implies lower recall).

The implemented systems that extract protein–concept associations show different degrees of complexity, ranging from simple cooccurrence of gene- and concept-indexed sentences to complex pattern matching and machine learning techniques. In addition, the unit chosen as context for the associations may vary – some use single sentences, others use passages or whole abstracts.

The simplest approach is concerned with cooccurrence extraction, i.e. mining those text fragments in which gene products and functional terms cooccur [47]. The Gene Information System (GIS) selects sentences containing functional information by indexing abstracts with genes and functional terms contained in a domain-specific lexicon (lexicon analysis). The iHOP [40] applications offer the possibility of extracting sentences with genes and functional terms such as GO or UMLS terms (Figure 3).

Most of the term-association extraction tools derive their functional concept lexicon from GO [5] or UMLS [11], or use Swiss-Prot keywords. GO concepts are principally constructed for annotating gene products unambiguously and consistently, regardless of their use for NLP tasks. Therefore, only a fraction of these terms can be mapped directly to biomedical abstracts [65]. Krallinger and coworkers [54] analyzed the compositional structure of GO terms to construct a rule-based system which generates natural language variants for certain GO terms. They also considered the individual word tokens that build-up these terms to retrieve annotation-relevant text passages.



Symbol	Name	Synonyms	Organism
WNT1	Wnt-1 proto-oncogene protein precursor	INT1	Homo sapiens
UniProt	P04628		
OMIM	164820		
NCBI Gene	7471		
NCBI RefSeq	NP_005421		
NCBI Accession	CAA26674, X03072		

Homologues of WNT1 ... new

Definitions for WNT1 ...

Enhanced PubMed/Google query ... new

WARNING: Please keep in mind that gene detection is done automatically and can exhibit a certain error. Read more

However, mAkt could act synergistically with Wnt-1 or Frat to **activate** LEF-1.

Beta-catenin: a common **target** for the regulation of cell adhesion by Wnt-1 and Src signaling pathways.

Wnt-1 **regulates** Fgf8 expression in the adjacent metencephalon, most likely via a secondary mesencephalic signal.

Cultured cells transfected with a membrane-tethered form of Wnt-1 **bind** epitope-tagged Frzb-1 in the 10(-10) M range.

In mammalian cells, Axin **inhibits** Wnt-1 stimulation of beta-catenin/lymphoid enhancer factor 1-dependent transcription.

Furthermore, beta-catenin is the **target** of two signal transduction pathways mediated by the proto-oncogenes src and wnt-1.

Ectopic expression of Wnt-1 in 3T3-L1 preadipocytes **stabilizes** beta-catenin, activates TCF-dependent gene transcription, and blocks adipogenesis.

Wrch1 gene is a down-stream **target** gene of Wnt1 in C57MG cells, and encodes a Cdc42-related GTPase with the potential to activate the JNK pathway.

Wnt-1 **induces** morphological transformation of C57MG mammary epithelial cells and accumulation of cytosolic beta-catenin whereas Wnt-5a has no effect.

On the contrary, coexpressed Wnt-1 and Frat **activated** LEF-1 but did not show significant inhibition of GSK-mediated phosphorylation of a peptide substrate.

The specificity of the approach enabled us to identify an Max-1 consensus DNA site within the transcriptional **control** region of the developmental regulatory gene Wnt-1.

FrzA efficiently **inhibited** the Wnt-1 mediated increase in cytoplasmic (beta)-catenin levels as well as the Wnt-1 induction of transcription from a Lef/tcf reporter gene.

Furthermore, a similar phenotype is not observed in Wnt1/RCAS-infected brains, demonstrating that ectopic Wnt1 is insufficient to mediate the **effect** of ectopic Lmx1b in our assay.

Figure 3 iHOP. The iHOP application allows for the automatic extraction of annotation-relevant sentences by identifying links between proteins and functional terms.

GoPubMed is a useful online resource which carries out an ontology-based literature search for a given PubMed query (e.g. using gene names) [26]. It utilizes a GO term extraction method based on morphological features (word stemming) and alignment of words forming GO terms to words which build-up PubMed abstracts. The terms forming the actual query as well as GO terms are highlighted in the retrieved abstracts and for each GO term an accuracy value as well as the definition of the term is provided.

Koike and coworkers followed a more NLP-based approach. They describe the association between genes/proteins/protein families and the functional concepts in the form of an *Actor* (which performs the action) and *Object* (which receives the action) relationship. To extract these relationships, they used shallow parsing and sentence structure analysis techniques. Terms which are semantically related to GO concepts are semiautomatically generated through literature-based term cooccurrence analysis and rules that result in morphological and syntactic variations of GO concepts.

The Medical Knowledge Explorer (MeKE) tool uses a different approach to creating synonyms for GO terms. It applies stemming and flexible matching before indexing the word tokens forming GO terms in the corresponding PubMed sentences. Then it extracts new GO synonyms by calculating the *edit distance* between the candidate synonyms and the actual GO terms. Via alignment of words forming sentences, MeKE also learns sentence motifs (consecutive words) relevant to functional descriptions. Finally, a naive Bayes classifier is used to estimate the overall likelihood of a given sentence corresponding to a gene–product–function description.

A statistical analysis of words appearing in Medline records is the basis of the GO engine [104], which calculates the frequency of association between terms derived from the literature and distinct GO concepts. The GO engine combines this text analysis with sequence similarity information and protein clustering to annotate proteins automatically.

A machine learning approach to the annotation of proteins with GO concepts based on whole abstracts was explored in Ref. [80]. They classified the documents associated with a given query protein according to their association with GO concepts by using a maximum entropy algorithm. The query protein is then annotated by combining the GO classifications from all the documents by means of a weighted voting scheme.

Regular expressions are often used to map biological concepts to the literature. The Textpresso system [68] consists of a text mining tool available online developed to support the process of WormBase database curation. It also operates on full-text articles, and includes classes of biological concepts and relations that include genes, alleles, cells and phenotypes as well as GO concepts. Additionally, classes which relate objects to each other (e.g. associations) are used. To tag these concepts, an extensive list of regular expressions was integrated into Textpresso.

Many of the tools which extract functionally relevant sentences or associations of proteins to functional terms are at an early stage and constitute recent developments, and therefore need further evaluation by the user community.

7.3 Mining Interactions and Relations from Text

Proteins instantiate their biological function by interacting with other biomolecules in the context of biological systems. For instance, enzymes which function as protein kinases catalyze the transfer of phosphate from ATP to the hydroxyl side-chains of the interacting proteins, and are part of important signaling pathways. The assembly of interaction networks of single proteins into a complex pathway is crucial to generalize function for a set of proteins (biological process) and to understanding the mode of operation of whole biological systems.

Large-scale techniques such as the yeast two-hybrid system or protein pull-downs are used to experimentally identify protein interactions. The biomedical literature holds information on both interaction partners and interaction types. II and IR methods are used to automatically identify these interactions from articles. One of the main advantages of text mining techniques with respect to the high-throughput experimental approaches (i.e. yeast two-hybrid or pull-down approaches; see Chapter 31) is their capacity of characterizing the nature of the interactions and their directionality, and not only the pure presence of an interaction. Interestingly, the structure of the interaction networks derived from the literature is similar to the one determined experimentally and both follow a characteristic power-law distribution with a scale-free topology [18, 42].

The extraction of protein interactions has attracted special attention within the biomedical text mining community. It is currently one of the most popular topics, and several online applications devoted to protein interaction extraction are currently available.

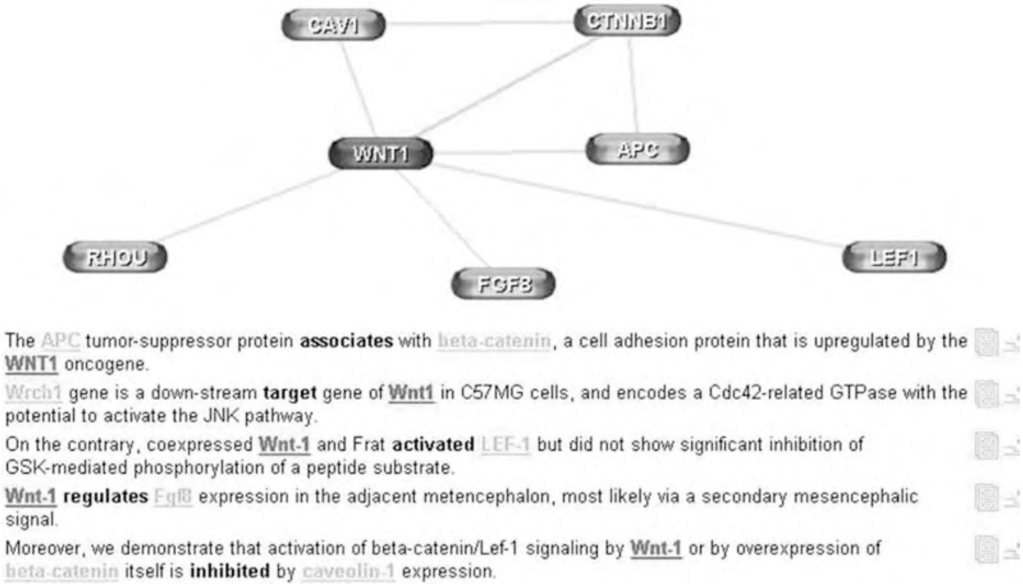
A straightforward strategy for extracting protein interactions is based on simple cooccurrence of previous gene- and protein-indexed articles. These approaches assume that, if two proteins frequently appear together in documents, they display a biological relationship. The PubGene system exploits the cooccurrence idea by indexing PubMed abstracts and titles with human proteins, and then constructing an interaction network based on all the binary interactions retrieved between cooccurring proteins [47].

Co-occurrence analysis alone does not permit the type of relationship between the interacting proteins to be determined. This was addressed by the SUISEKI [10] system, which is based on the use of frames, i.e. patterns which correspond to language expressions used to describe protein interactions within single sentences. Each interaction is scored on the basis of the frequency of repetition of cooccurring proteins within frames as well as on the individual reliability score calculated for each frame type.

The Internet-based Chilibot application [18] includes NLP methods such as POS tagging and shallow parsing, as well as a set of rules to extract relationships between biological entities, i.e. genes, proteins and drugs. This tool defines the directionality of interactions and classifies sentences into five basic types: in addition to abstract cooccurrence, it defines stimulatory, inhibitory and neutral interactions, as well as negative interactions (noninteractions).

The Biomolecular Interaction Database (BIND) contains information about curated molecular interactions. Extracting these interactions manually from the literature is very time consuming. To speed-up the discovery process, a machine learning technology called PreBIND [27], based on SVMs, was developed. This online tool is able to classify whether articles describe biomolecular interactions.

iHOP consists of an application that automatically links proteins detected through their cooccurrence in the literature. It offers an easy-to-use web interface that highlights in text functional terms and interaction patterns of proteins, allowing navigation between concepts by jumping from protein to protein in hyperlinked space. The interaction network for a given protein query is visualized as a graph and allows for navigation through the associ-



The APC tumor-suppressor protein **associates** with beta-catenin, a cell adhesion protein that is upregulated by the **WNT1** oncogene.

Wrch1 gene is a down-stream **target** gene of **Wnt1** in C57MG cells, and encodes a Cdc42-related GTPase with the potential to activate the JNK pathway.

On the contrary, coexpressed **Wnt-1** and Frat **activated** LEF-1 but did not show significant inhibition of GSK-mediated phosphorylation of a peptide substrate.

Wnt-1 **regulates** Fgf8 expression in the adjacent metencephalon, most likely via a secondary mesencephalic signal.

Moreover, we demonstrate that activation of beta-catenin/Lef-1 signaling by **Wnt-1** or by overexpression of beta-catenin itself is **inhibited** by caveolin-1 expression.

Figure 4 Interaction graph provided by iHOP.

ated proteins. Interactions supported by experimental evidence are ranked higher in the returned list of interaction sentences (Figure 4).

The Genomics Information Extraction System (GENIES) [84] uses a knowledge base to organize and structure information about molecular pathways derived from the literature, and also considers the extraction of complex nested expressions referring to interactions.

One of the main difficulties encountered in protein interaction extraction strategies has been the lack of large, well-curated training and test data sets consisting of sentences of text passages referring to protein interactions. Although a recent community-wide evaluation addressed this issue [70], many tools are evaluated using information provided by interaction databases such as DIP [103]. These databases contain well-curated protein interactions generally extracted by domain experts from full-text articles [18]. As protein interaction extraction tools in general only have access to abstracts rather than full-text articles, many of the interactions contained in other sections of the articles are missed. Also, the prior protein-tagging step is crucial to identify potential protein interactions [10].

A commonly accepted formalization of interaction types which can be derived from the literature is missing. Therefore different tools define interactions (relations) at different degrees. Some consider interactions at a rather coarse level [44], focusing more on extraction of relationships between proteins than on extraction of physical protein interactions. Those tools are able to also extract relationships based, for instance, on homology, e.g. the mention of to homologous genes from different organism. Other methods are more centered on extraction of very specific interaction types, e.g. inhibitory or stimulatory relations [18] or even phosphorylation [10]. Even if the different existing online approaches provide complementary information to each other, it is still far from possible to connect them in useful workflows. Although the most common interactions studied are those between proteins, some effort has been done for the analysis of the interactions between proteins and drugs.

7.4 Discovering Information Associated with Groups of Proteins

The coordinated expression of a certain gene within a group of genes studied in microarray experiments provides important information about coregulation within biological processes (for a detailed description of microarray technologies, see Chapter 24). Text mining tools have been developed to assist in the interpretation of microarray data. These techniques characterize groups of genes by extracting words with functional meaning that are statistically associated to the corresponding literature [8] and use this information to score the coherence of gene clusters [81]. Moreover, text mining techniques are used to complement the description of the relationships between genes within gene clusters obtained through the similarity of their expression patterns by, for example, pointing out genes that are coexpressed, but do not share a known biological function. These genes can be detected by the associated functional terms extracted from the literature) and can therefore correspond to the discovery of a new biological association. Jenssen and coworkers [47] superimposed gene expression data on a literature-based network, detecting gene relationships which were not previously identified by clustering techniques. Another study showed that, in general, there is a correlation between the similarity of the expression patterns and the significance of functional information derived from the literature [75]. The GEISHA system [8] uses terms extracted from the literature with a given significance value to characterize the functions potentially associated to each cluster of genes. GEISHA additionally uses double-words and sentences to increase the interpretability of the extracted information in biological terms.

The ConceptMarker algorithm has been applied to integrate analysis of gene expression data associated to juvenile arthritis with biomedical literature analysis [59]. It uses the ProMiner protein tagger to the detect genes in abstracts – mapping the genes studied in the microarray experiment with articles mentioning them [34]. To aid in the interpretation of information associated to the experimental gene clusters it identifies a set of literature terms, which describe consistently these clusters using VSM-based term weighting and singular value decomposition (SVD) methods.

A number of systems have been developed that instead of using literature mining directly, exploit associations of genes to GO terms provided in annotations databases like GOA. The corresponding terms are used in ways that are similar to the above methods to characterize functions common to genes with similar expression profiles and to discover new potential associations. An example of this type of systems is FatiGO, that uses statistical tests to identify relevant GO terms for groups of genes based on GOA associations [2].

An advantage of approaches based on IE methods are that the amount of functional information contained in the scientific articles is clearly more complete than incorporated in databases and ontologies, and they can also benefit from a richer expressivity regarding functional characteristics. In contrast, the GO-based approaches are easier to implement and have very simple interpretations since they use a controlled vocabulary.

Even if it is clear that both literature- and GO-based approaches have advantages that can be complementary, a full documented study of their possible combination has not yet been performed.

7.5 Other Applications

Although the majority of IE and IR tools for life science literature are concerned with the identification of biological entities, interactions, functional descriptions and functional terms describing gene clusters, other biological problems have also been considered.

Chromosome aberrations are known to be associated in many cases to human pathologies and cancer conditions. To extract the corresponding breakpoint manually from the literature is a labor-intensive task. The HCAD database contains information on human genomics breakpoints and chromosome aberrations automatically extracted from the literature [40].

For the detection of genomic variations, especially in the context of cancer-specific texts, a tool named VTag based on machine learning methods is available [66], with an f-score of 0.8192.

The regulation of gene function in higher organisms is subjected to tissue-specific control mechanisms. In fact, transcript diversity is especially important for tissue-specific gene expression. NLP methods have been developed to identify alternative transcripts of genes detected in PubMed [87].

Although most gene and protein sequences are stored in biological databases, short sequence patterns with significant functional properties like binding sites or epitopes are frequently mentioned directly in texts. An information extraction method based on Markov models to locate such sequence patterns in large text collections has been implemented. For the identification of peptide epitopes, this system is able to reach a considerable level of precision (precision of 67% with a recall of 85%) [102].

The efficiency of reactions carried out by enzymes can be expressed through kinetic parameters. Although databases such as the Kyoto Encyclopedia of Genes and Genomes (KEGG) contain extensive descriptions about enzymes, existing information about kinetic aspects is still often hidden within articles. A text classification system based on SVMs tackles this problem and assists researchers in the search for information relevant to kinetic aspects of enzymes [33].

Eukaryotic cells compartimentalize certain biological processes. The subcellular location of proteins gives additional clues about certain functional aspects, e.g. transcription factors display their molecular function in the nucleus. Not only bioinformatics methods are able to classify protein sequences according to their subcellular location, but also literature associated to a given protein is useful to infer its subcellular location [69, 90].

Protein sequence similarity searches are the base for automatically annotating protein function or finding suitable templates for protein structure prediction methods. To determine if two proteins are remotely homologs in cases where sequence similarity is in the twilight zone often requires human intervention and manual consultation of annotation database records. The combination of sequence search methods such as PSI-BLAST with standard text similarity method such as the VSM has been used in the SAWTED system to enhance remote homology searches [62, 94].

8 Evaluation of Text Mining Strategies

The importance of community-wide evaluations has been realized by both the bioinformatics and NLP communities. In the case of the bioinformatics field, open evaluations have been carried out for tasks such as protein structure prediction microarray data analysis [48] and gene finding [83]. Evaluations of NLP strategies for IE systems (MUC) and IR end-document classification methods have also been performed for several years [39, 95].

Each of the first IR, IE and text mining tools developed for the biomedical domain used their own evaluation data sets and strategies, which hindered performance comparison and determination of the state art. The construction of suitable textual test and training data requires involvement of domain experts and is labour intensive. Therefore, many systems evaluated performance using small data sets that were not sufficiently representative.

"Gold standard" data sets provided by human experts are used to consistently score performance of different text mining systems. The agreement of these experts is evaluated by means of inter-annotator agreement. The lower this agreement, the more difficult the task. In order to ensure objective evaluation metrics, comparing cross-system performance and producing "gold standard" data sets community-wide evaluations are crucial.

The retrieval of documents with relevant information for a given biological problem is the first step in the processing of text collections. To assess document categorization and retrieval systems in the biomedical domain was the main goal of the TREC Genomics [37] track. In the case of the TREC Genomics track 2004, one of the main challenges was related to retrieval of PubMed abstracts relevant to previously defined topics. These topics consisted basically of information needs formulated by biologists and which should reflect common information demands encountered in this domain. The average precision of the top-scoring systems was around 43% of the retrieved articles containing correct information. The second task of TREC 2004 was organized into subtasks. One of them, the triage subtask, focused on the classification of full-text articles regarding the presence of experimental evidence useful for database curators to extract annotations. The annotation subtask had the goals of identifying, given an article and gene name, which GO categories (i.e. molecular function, biological process and cellular component) were used in the articles to annotate gene products. These two subtasks actually reflect some of the initial steps undertaken by database curators to derive documents relevant for further manual curation.

A similar task was posed at the The KDD challenge cup [106], concerned with the retrieval of papers containing experimental evidence for a given set of *Drosophila* gene products (relevant to extract annotations).

The task of indentifying entities (gene and protein names) was addressed by the BioCreative comunity [105]. A total of 15 teams participated in this experiment (subtask 1A), trying to locate precisely gene names within full-text articles provided by the annotators as valid references for those genes. Top-scoring groups reached an f-score (balanced precision and recall) of 0.8, which is a promising result, but still far below the 0.9 f-score for detection of other names such as company names of locations tagged in newswire texts.

A similar challenge was held at the shared task of the JNLPBA [50]. In this contest the recognition of different biological entities included protein names, DNA, RNA, cell line and cell type. The introduction of additional entities decreased the overall performance when compared to BioCreative task 1A, reaching an f-score of 0.76. The ABNER protein tagger (f-score of 0.703) [86] and a system based on PowerbioNE were among the top scoring strategies [107].

For biological applications the tagging of protein names in text is not sufficient and it is additionally necessary to link them to the corresponding sequence database entries. Subtask 1B of the BioCreative challenge [38] addressed this point, requiring the identification of protein/gene names and their corresponding database links in examples from three model organisms:

yeast, fly and mouse. Among the best-performing systems was the rule-based ProMiner approach [34]. The differences between the results in the three organisms were interesting – whereas yeast genes were rather easy to identify (f-score of 0.92), fly genes (f-score of 0.82) and mouse genes (f-score of 0.79) were more complicated to link to database entries. An explanation for this difference could be that yeast genes are commonly mentioned in articles using short names or well-defined unambiguous symbols, whereas fly and mouse genes are mentioned often using ambiguous or long gene names.

Many applications relating to the recovery of protein interactions from the literature were implemented. The Genic Interaction Extraction Challenge [70] supplied a training and test set to benchmark the state of the art in text-based protein interaction extraction. In this case, the best-performing teams could reach an f-score of 0.518.

In the KDD cup and Genomics TREC contests, articles which contain relevant information for curators are retrieved, but the annotations and annotation-relevant text passages must be identified manually by domain experts. The second task of the BioCreative community addressed a more advanced step in text mining – the automatic identification of functional annotations associated to genes and proteins [7]. In subtask 2A, a protein identifier (Swiss-Prot Id), a GO concept and a full-text article selected by the GOA annotators as representative of the protein–GO association were provided to the participants, and they were asked to highlight significant passages of the text demonstrating the protein–GO relation. In this case, in addition to the difficulty of tagging gene names common to task 1, references to the GO terms had to be identified, and both had to be correlated within the same passage – a difficulty of considerable magnitude given the large semantic variability in which GO terms can be expressed in texts. Top-scoring systems submitting results for all the posed queries reached a precision of 0.28, while participants only submitting results for a small number of high-confidence cases reached a precision of 0.8. Subtask 2B was even more challenging as only protein–full-text article pairs were provided and participants were asked to predict the GO term useful for annotating the proteins together with the text passage containing the evidence supporting the annotation.

9 Resources for Text Mining

A range of resources is available to develop text mining and NLP applications suitable to extract information from biology literature, including text repositories and databases, annotated text corpora for training and testing purposes, gene dictionaries as well as generic and domain specific NLP tools. Table 2 gives a detailed list of resources and resource types.

9.1 Literature Databases

The biomedical literature and especially abstracts from the PubMed database constitute the primary data resource. Interestingly systems developed for processing generic texts are increasingly being adapted to biomedical text processing. A more detailed of description of existing biomedical literature databases is provided in Section 3.

9.2 Annotated Text Corpora

Tagged biomedical text corpora are essential to train text classifiers, machine learning techniques and to extract text patterns. Although there is increasing interest in producing open access corpora for biomedical text mining applications, only a few large data sets are currently available.

The GENIA corpus [49] consists of a collection of biomedical abstracts which have been semantically annotated and which has been used as a benchmark set for biology NER tools [50]. The annotation of biological entries is based on a predefined ontology (GENIA ontology). GENIA Corpus Version 3.0x consists of 2000 abstracts related to human, blood cells, and transcription factors.

The BioCreative challenge resulted in data sets useful for tagging protein and genes in articles to protein database entries. Additionally, it produced a data set of text passages describing protein–GO associations evaluated by GOA database curators as positives and negatives (Blaschke et al.).

Another data set used for protein entity tagging is the Yapex corpus, which has been used to test the NLProt tool [28]. The Genic Interaction Extraction Challenge has provided both a training and test set for protein-protein interaction discovery from text [70].

A list of other text data sets useful for different text mining task is contained in Table 2.

9.3 Generic NLP Tools

A vast amount of existing online tools and software can be found for processing generic literature or newswire texts. It is beyond the scope of this chapter to provide a complete list of different applications and only a small number of systems will be mentioned. In principle, when developing NLP tools for biomedical literature, existing software for generic texts could be adapted.

Table 2 Tools and databases for biomedical text mining

Type	Name	Web site	Availability[a]
Literature databases	PubMed [98]	www.ncbi.nlm.nih.gov/PubMed	D,W
Annotated text corpora	Biocreative [38, 105] [7]	www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/results	D
	GENIA corpus [49]	www-tsujii.is.s.u-tokyo.ac.jp/ genia/topics/Corpus/	D
	Yapex corpus [28]	www.sics.se/humle/projects/prothalt/#data	D
	PASBio [96]	research.nii.ac.jp/ collier/projects/PASBio	D
	LLL05 dataset [70]	genome.jouy.inra.fr/texte/LLLchallenge/#task1	D
	Medstract corpus [79]	www.medstract.org/gold-standards.html	D
	FetchProt corpus	fetchprot.sics.se	D
Generic NLP tools	Stanford Lexical Parser [51]	nlp.stanford.edu/software/lex-parser.shtml	S
	Nice stemmer	ils.unc.edu/iris/irisnstem.htm	W
	Bow [64]	www.cs.cmu.edu/ mccallum/bow	S
	GATE [12]	gate.ac.uk	S
	NLTK [61]	nltk.sourceforge.net/index.html	S
	CCG tools	l2r.cs.uiuc.edu/ cogcomp/tools.php	S
Dictionaries and ontologies	GO/OBO [5]	obo.sourceforge.net	D
	UMLS [11]	www.nlm.nih.gov/research/umls/umlsmain.html	D
	MeH [71]	www.nlm.nih.gov/mesh/meshhome.html	D
Biomedical NLP tools	NLProt [67]	cubic.bioc.columbia.edu/services/nlprot	S
	AbGene [92]	ftp://ftp.ncbi.nlm.nih.gov/pub/tanabe/AbGene	S
	ABNER [86]	www.cs.wisc.edu/%7Ebsettles/abner	S
	PowerbioNE [107]	textmining.i2r.a-star.edu.sg/NLS/webdemo/bioner.html	W
	iHOP [40]	www.pdg.cnb.uam.es/UniPub/iHOP	W
	EBIMed	www.ebi.ac.uk/Rebholz-srv/ebimed/index.jsp	W
	eTBLAST: [97]	invention.swmed.edu/etblast/index.shtml	W
	GoPubMed [26]	www.gopubmed.org	W
	iProLINK [45]	pir.georgetown.edu/iprolink	W
	METIS [108]	umber.sbs.man.ac.uk/cgi-bin/dbbrowser/precis/metis_precis.cgi	W
	PreBIND [27]	www.blueprint.org/products/prebind/prebind_search.html	W
	Chilibot [18]	www.chilibot.net	W
	Textpresso [68]	www.textpresso.org	W
	XplorMed [77]	www.ogic.ca/projects/xplormed	W
	MedPost [88]	3did.embl.de	W

∗: Software S, Web site/server W, Data set D.

For statistical text analysis, the Bow Toolkit is very useful [64]. It is written in the C programming language, and provides libraries for text retrieval (arrow), classification (rainbow), and clustering (crossbow) tasks.

Generic tools for analyzing all the different features of natural language are available. To explore the grammatical and syntactical characteristics of language, toolkits such as GATE [12], NLTK and CCG integrate many text-processing applications such as POS tagging and parsing [61].

Morphological features (e.g. stripping off suffixes of words) can be analyzed by stemmer algorithms and are a useful step in IR. Among the existing stemming algorithms for English texts are the Porter stemmer [78], which carries out a rather deep analysis, and the Krovetz system, which can be mentioned as an example of "light" stemming [58].

9.4 Dictionaries and Ontologies

Collections of gene and protein names, symbols and synonyms stored in biological databases such as Swiss-Prot are valuable resources, providing the base for name identification and linking of text to database identifiers. Thesauri and ontologies are also essential for building the background knowledge in which to map the information extracted from text. In the biomedical domain, the two larger collections are GO and UMLS. w

GO consists of a controlled vocabulary of concepts which are relevant to biological aspects such as molecular function, biological process and cellular location [5]. GO is part of a repository of ontologies, called Open Biomedical Ontologies (OBO), which contains additional ontologies such as the Human Disease Ontology and the Sequence Ontology. GO contains more than 16 000 concepts, and for some of them synonyms and definitions are also provided. There are over 200 000 annotations of gene products using GO terms, contained in annotation databases such as GOA (see also Chapter 29).

Although GO is currently the most widespread vocabulary resource for annotation purposes, it lacks ways of quantifying similarities between concepts. Also, the relationships between terms are mainly restricted to "is a" and "part of" relations. Finally, for text mining purposes, many terms have a limited use because they do not correspond to language constructs or expressions used in the literature.

The UMLS Metathesaurus [11] contains more than 1 million concepts in biomedicine and health science as well as the relationships between them. As it focuses on medical terms rather than molecular biology vocabulary, it has been shown to be more useful for medical than for biology text mining. Finally, the MeSH thesaurus provides a set of controlled vocabulary terms organized in a hierarchical structure which is used to index PubMed records [71]. It contains more than 22 000 terms ranging from very broad ones such as

"Anatomy" to more specific ones like "Ankle". As in the case of the UMLS, MeSH is tailored towards medical literature rather than biology.

9.5 Biomedical Domain NLP Systems

A considerable number of biomedical NLP applications have already been mentioned throughout this chapter. Some of them are useful for very specific tasks, such as the MedPost [88] tool – a POS tagger adapted to biomedical texts. A range of protein name taggers is available either as a downloadable software or as online servers. The NLProt tagger, a SVM-based tagger, also links the identified proteins to database entries [67], while the ABNER tagger identifies additionally cell lines, RNA and DNA.

Providing graphical representation of information useful for the human interpretation of knowledge is an essential challenge for the text mining field and will ultimately influence its perception in the large community of potential users. iHOP facilitates the understanding of extracted protein relationships by highlighting the identified entities, allowing a graph representation of the interaction network, and facilitating the navigation between different concepts and databases [40]. Other systems that provide comprehensive graphical web interfaces are EBIMed and Chilibot [18] (Table 2).

10 Concluding Remarks

Processing of scientific text is a challenging task due to the complexity of human language. Nonetheless, there is growing interest in text mining and IE technology applied to biomedical literature as a considerable fraction of existing biological information is only available in the literature. This technology will play an important role in the exploration of high-throughput experiments, in the annotation of biological databases and in the analysis of complex biological problems.

The biological database community was perhaps the first with defined expectations towards the development of NLP tools which could assist human experts in accessing relevant functional descriptions or helping in the extraction of relevant annotations. Collaborations of databases including Mouse Genome Informatics (MGI), FlyBase, Swiss-Prot, NCBI and annotation databases like GOA with biomedical text mining groups have resulted in fruitful community-wide evaluations where practical tasks were posed. The GO consortium has even considered text-based computation as one of the strategies for its annotation process. There are already some examples of text mining tools assisting biological databases, as is the case of iHOP, and the interaction database IntAct [36] or PreBIND and the BIND database [27]. Integration of NLP tools will increase efficiency of data acquisition and traceability in the future.

Still, the construction of suitable data sets for the development and assessment of NLP systems in this domain will require further efforts of collaboration with databases and biology domain experts. A particularly crucial aspect for the development of this field would be the availability of domain ontologies and thesauri. The rapid advance of technologies in molecular biology and biomedicine is producing large and complex collections of data, for which we essentially lack the appropriate formalized structures and terminologies. All these developments have a clear incidence in text mining technologies that use structured field knowledge as platform. Finally, the assessments of NLP strategies applied to biology literature have been revealed as a powerful natural way of creating community, focusing on key biological problems and fostering developments. They also provide a comprehensive view of the status of the field.

The first NLP systems applied to molecular biology and biomedicine were initially developed 10 years ago, and centered on identifying protein names and interactions. Meanwhile, they have diversified to larger a set of applications. This diversification was partially promoted by the fact that very specific functional information was missing in many of the existing databases, but was relevant to the needs of different areas of biomedicine and molecular biology. These NLP developments included extraction of information about kinetic parameters, chromosome aberrations or alternative transcripts.

Although PubMed is the central literature repository for biologists, it does not facilitate specific retrieval or extraction of gene-related information. When querying PubMed, it is necessary to browse through large lists of abstracts until information of interest is detected. New systems like iHOP or Chilibot allow more efficient access and visualization of gene-relevant textual data, and constitute alternatives to traditional PubMed searches.

The possibilities for complementing traditional bioinformatics approaches with NLP methods, creating hybrid systems, are still largely unexplored. Initial attempts have been restricted to enhance sequence searches with information extracted from abstracts. The development of new hybrid approaches integrating a larger diversity of data types such as diseases, phenotypic information or experimental conditions remains as a challenge for the years to come.

Acknowledgments

The work of our group was supported by grants of the European Commission (ENFIN LSH6-CT-2005-518254 , Biosapiens LSHC-CT-2003-505265). Thanks to Burr Settles for providing Figure 2, to Robert Hoffmann for providing Figures 3 and 4, to Boris Galitsky for additional information about his QA system, and to Ana Divoli and Thomas Lengauer for useful comments and suggestions.

References

1 ABNEY, S. 1991. *Parsing By Chunks. Principle-Based Parsing.* The MIT Parsing Volume, 1988–89, Center for Cognitive Science, MIT.

2 AL-SHAHROUR, F., R. DIAZ-URIARTE and J. DOPAZO. 2004. FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. Bioinformatics **20**: 578–80.

3 ANDRADE, M. AND A. VALENCIA. 1997. Automatic annotation for biological sequences by extraction of keywords from MEDLINE abstracts. Development of a prototype system. Proc. ISMB **5**: 25–32.

4 ANTONIOTTI, M., I. LAU AND B. MISHRA. 2004. Naturally speaking: a systems biology tool with natural language interfaces. TR2004: 853.

5 ASHBURNER, M., C. BALL, J. BLAKE et al. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. **25**: 25–9.

6 BAEZA-YATES, R. and B. RIBEIRO-NETO. 1999. *Modern Information Retrieval.* Addison Wesley, Reading, MA.

7 BLASCHKE, C., E. ANDRES LEON, M. KRALLINGER and A. VALENCIA. 2005. Evaluation of BioCreative assessment of task 2. BMC Bioinformatics **6**: S16.

8 BLASCHKE, C., J. OLIVEROS and A. VALENCIA. 2001. Mining functional information associated with expression arrays. Funct. Integr. Genomics. **1**: 256–68.

9 BLASCHKE, C., L. HIRSCHMAN and A. VALENCIA. 2002. Information extraction in molecular biology. Brief Bioinform. **3**: 154–65.

10 BLASCHKE, C. and A. VALENCIA. 2002. The frame-based module of the Suiseki information extraction system. IEEE Intell. Syst. **17**: 14–20.

11 BODENREIDER, O. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. Nucleic Acids Res. **32**: 267–70.

12 BONTCHEVA, K., V. TABLAN, D. MAYNARD and H. CUNNINGHAM. 2004. Evolving GATE to meet new challenges in language engineering natural language engineering. Natural Language Eng. **10**: 349–73.

13 BUTLER, D. 2004. Science searches shift up a gear as Google starts Scholar engine. Nature **432**: 423.

14 BUTLER, G. 2005. Workflow scenarios for a semantic web for fungal genomics. In Proc. NETTAB, Naples, Italy: 101–4.

15 CAMON, E., M. MAGRANE, D. BARRELL, et al. 2003. The Gene Ontology Annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, and InterPro. Genome Res. **13**: 662–72.

16 CASTANO, J., J. ZHANG and J. PUSTEJOVSKY. 2002. Anaphora resolution in biomedical literature. In Proc. Int. Symp. on Reference Resolution, Alicante, Spain.

17 CHANG, J., H. SCHUTZE and R. ALTMAN. 2004. GAPSCORE: finding gene and protein names one word at a time. Bioinformatics **20**: 216–25.

18 CHEN, H. and B. SHARP. 2004. Content-rich biological network constructed by mining PubMed abstracts. BMC Bioinformatics **5**: 147.

19 CHEN, L., H. LIU and C. FRIEDMAN. 2005. Gene name ambiguity of eukaryotic nomenclatures. Bioinformatics **21**: 248–56.

20 CHIANG, J., H. YU and H. HSU. 2004. GIS: a biomedical text-mining system for gene information discovery. Bioinformatics **20**: 120–1.

21 COHEN, K. and L. HUNTER. 2005. Natural language processing and systems biology. In Dubitzky and Pereira, Artifical Intelligence Methods and Tools for Systems Biology, Springer-Verlag, Heidelberg, Germany.

22 CORNEY, D., B. F. BUXTON, W. LANGDON and D. JONES. 2004. BioRAT: extracting biological information from full-length papers. Bioinformatics **20**: 3206–13.

23 DAGAN, I., B. MAGNINI and O. GLICKMAN. 2005. The PASCAL recognizing textual entailment challenge. In Proc. First Recognizing Textual Entailment Workshop, Southampton, UK: 1–9.

24 DALE, R., E. HOVY, D. R¨OSNER and O. STOCK. 1992. *Aspects of Automated Natural Language Generation*. Springer, Heidelberg.

25 DIVOLI, A. and T. ATTWOOD. 2005. BioIE: extracting informative sentences from the biomedical literature. Bioinformatics **21**: 2138–9.

26 DOMS, A. and M. SCHROEDER. 2005. GoPubMed: exploring PubMed with the Gene Ontology. Nucleic Acids Res. **33**: W783–6.

27 DONALDSON, I., J. MARTIN, B. DEBRUIJN, et al. 2003. PreBIND and Textomy – mining the biomedical literature for protein-protein interactions using a support vector machine. BMC Bioinformatics **4**: 11.

28 FRANZEN, K., G. ERIKSSON, F. OLSSON, L. ASKER, P. LIDEN and J. COSTER. 2002. Protein names and how to find them. Int. J. Med. Inf. **67**: 49–61.

29 FUKUDA, K., T. TSUNODA, A. TAMURA and T. TAKAGI. 1998. Toward information extraction: identifying protein names from biological papers. Pac. Symp. Biocomput. **3**: 707–18.

30 GALITSKY, B. 2001. A natural language question-answering system for Human Genome domain. 2nd IEEE Intl Symposium on Bioinformatics and Bioengineering, Rockville, MD, USA.

31 GALITSKY, B. 2003. *Natural Language Question Answering System.* Advanced Knowledge International, Adelaide.

32 GENE ONTOLOGY CONSORTIUM. 2004. The Gene Ontology (GO) database and informatics resource. Nucleic Acids Res. **32**: D258–61.

33 HAKENBERG, J., S. SCHMEIER, A. KOWALD, E. KLIPP and U. LESER. 2004. Finding kinetic parameters using text mining. OMICS **8**: 131–52.

34 HANISCH, D., K. FUNDEL, H. MEVISSEN, R. ZIMMER and J. FLUCK. 2005. ProMiner: rule-based protein and gene entity recognition. BMC Bioinformatics **6**: S14.

35 HAUSSER, R. 1999. *Foundations of Computational Linguistics: Human-Computer Communication in Natural Language*. Springer, Berlin.

36 HERMJAKOB, H., L. MONTECCHI-PALAZZI, C. LEWINGTON, et al. 2004. IntAct: an open source molecular interaction database. Nucleic Acids Res. **32**: D452–5.

37 HERSH, W., R. BHUPATIRAJU, L. ROSS, P. JOHNSON, A. COHEN and D. KRAEMER. 2004. TREC 2004 Genomics Track Overview. In Proc. Text Retrieval Conf. Gaithersburg, MD, USA.

38 HIRSCHMAN, L., M. COLOSIMO, A. MORGAN and A. YEH. 2005. Overview of BioCre-AtIvE task 1B: normalized gene lists. BMC Bioinformatics **6**: S11.

39 HIRSCHMAN, L. 1998. The evolution of evaluation: lessons from the message understanding conference. Comput. Speech Language **12**: 281–305.

40 HOFFMANN, R., J. DOPAZO, J. CIGUDOSA and A. VALENCIA. 2005. HCAD, closing the gap between breakpoints and genes. Nucleic Acids Res. **33**: D511–3.

41 HOFFMANN, R., M. KRALLINGER, E. ANDRES, J. TAMAMES, C. BLASCHKE and A. VALENCIA. 2005. Text mining for metabolic pathways, signaling cascades, and protein networks. Sci. STKE **283**: pe21.

42 HOFFMANN, R. and A. VALENCIA. 2003. Protein interaction: same network, different hubs. Trends Genet. **19**: 681–3.

43 HOFFMANN, R. and A. VALENCIA. 2004. A gene network for navigating the literature. Nat Genet. **36**: 664.

44 HOFFMANN, R. and A. VALENCIA. 2005. Implementing the iHOP concept for navigation of biomedical literature. Bioinformatics **21**: ii252–8.

45 HU, Z., I. MANI, V. HERMOSO, H. LIU and C. WU. 2004. iProLINK: an integrated protein resource for literature mining. Comput. Biol. Chem. **25**: 409–16.

46 HU, Z., M. NARAYANASWAMY, K. RAVIKUMAR, K. VIJAY-SHANKER and C. WU. 2005. Literature mining and database annotation of protein phosphorylation using a rule-based system. Bioinformatics **21**: 2759–65.

47 JENSSEN, T., A. LAEGREID, J. KOMOROWSKI and E. HOVIG. 2001. A literature network of human genes for high-throughput analysis of gene expression. Nat. Genet. **28**: 21–8.

48 JOHNSON, K. and S. LIN. 2001. Critical assessment of microarray data analysis: the 2001 challenge. Bioinformatics **17**: 857–8.

48a KANG, C.-G. and J. C. PARK. 2005. Generation of coherent gene summary with concept-linking sentences. Proc. Int. Symp. Languages in Biology and Medicine (LBM). Daejon, Korea: 41–5.

49 KIM, J., T. OHTA, Y. TATEISI and J. TSUJII. 2003. GENIA corpus – semantically annotated corpus for bio-textmining. Bioinformatics **19**: i180–2.

50 KIM, J., T. OHTA, Y. TSURUOKA and Y. TATEISI. 2004. Introduction to the Bio-Entity recognition task at JNLPBA. Proc. JNPBA: 70–6.

51 KLEIN, D. and C. MANNING. 2002. Fast exact inference with a factored model for natural language parsing. Advances in NIPS 2002 Westin, Canada.

52 KOIKE, A., Y. NIWA and T. TAKAGI. 2005. Automatic extraction of gene/protein biological functions from biomedical text. Bioinformatics **21**: 1227–36.

53 KOSTOFF, R., J. BLOCK, J. STUMP and K. PFEIL. 2004. Information content in Medline record fields. Int. J. Med. Inform. **73**: 515–27.

54 KRALLINGER, M., M. PADRON and A. VALENCIA. 2005. A sentence sliding window approach to extract protein annotations from biomedical articles. BMC Bioinformatics **6**: S19.

55 KRALLINGER, M., M. PADRON, C. BLASCHKE and A. VALENCIA. 2004. Assessing the correlation between contextual patterns and biological entity tagging. Proc. In NLPBA/COLING, Geneva, Switzerland: 36–43.

56 KRALLINGER, M., R. ALONSO-ALLENDE and A. VALENCIA. 2005. Text-mining approaches in molecular biology and biomedicine. Drug Discov. Today **10**: 439–45.

57 KRAUTHAMMER, M., A. RZHETSKY, P. MOROZOV and C. FRIEDMAN. 2000. Using BLAST for identifying gene and protein names in journal articles. Gene **259**: 245–52.

58 KROVETZ, R. 1993. Viewing morphology as an inference process. In Proc. 16th ACM SIGIR Conf., Pittsburgh, PA, USA: 191–202.

59 KUFFNER, R., K. FUNDEL and R. ZIMMER. 2005. Expert knowledge without the expert: integrated analysis of gene expression and literature to derive active functional contexts. Bioinformatics **21**: ii259–67.

60 LEE, D., H. CHUANG and K. SEAMONS. 1997. Document ranking and the vector-space model. IEEE Software **14**: 67–75.

61 LOPER, E. and S. BIRD. 2002. NLTK: the natural language toolkit. In Proc. ACL, Philadelphia, PA, USA.

62 MACCALLUM, R., L. KELLEY and M. STERNBERG. 2000. SAWTED: structure assignment with text description–enhanced detection of remote homologues with automated SWISS-PROT annotation comparisons. Bioinformatics **16**: 125–9.

63 MANNING, C. and H. SCHUETZE. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

64 MCCALLUM, A. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering.

65 MCCRAY, A., A. BROWNE and O. BODENREIDER. 2002. The lexical properties of the gene ontology. In Proc. AMIA Symp., San Antonio, Texas: 504–8.

66 MCDONALD, R., R.WINTERS, M. MANDEL, Y. JIN, P. WHITE and F. PEREIRA. 2004. An entity tagger for recognizing acquired genomic variations in cancer literature. Bioinformatics **20**: 3249–51.

67 MIKA, S. and B. ROST. 2004. NLProt: extracting protein names and sequences from papers. Nucleic Acids Res. **32**: W634–7.

68 MULLER, H., E. KENNY and P. STERNBERG. 2004. Textpresso: an ontology-based information retrieval and extraction system for biological literature. PLoS Biol. **2**: e309.

69 NAIR, R. and B. ROST. 2002. Inferring sub-cellular localization through automated lexical analysis. Bioinformatics **8**: S78–86.

70 NEDELLEC, C. 2005. Learning language in logic – genic interaction extraction challenge. In Proc. LLL05 Workshop, Bonn, Germany.

71 NELSON, S., M. SCHOPEN, A. SAVAGE, J. SCHULMAN and N. ARLUK. 2004. The MeSH translation maintenance system: structure, interface design, and implementation. In Proc. 11th World Congr. on Medical Informatics, San Francisco, USA: 67–9.

72 NENADIC, G., I. SPASIC and S. ANANIADOU. 2003. Terminology-driven mining of biomedical literature. Bioinformatics **19**: 938–43.

73 NETZEL, R., C. PEREZ-IRATXETA, P. BORK and M. ANDRADE. 2003. The way we write. EMBO Rep. **4**: 446–51.

74 NEUFELD, T. and G. RUBIN. 1994. The Drosophila peanut gene is required for cytokinesis and encodes a protein similar to yeast putative bud neck filament proteins. Cell **77**: 371–9.

75 OLIVEROS, J., C. BLASCHKE, J. HERRERO, J. DOPAZO and A. VALENCIA. 2000. Expression profiles and biological function. Genome Inform. Ser. Workshop Genome Inform. **11**: 106–17.

76 ONO, T., H. HISHIGAKI, A. TANIGAMI and T. TAKAGI. 2001. Automated extraction of information on protein–protein interactions from the biological literature. Bioinformatics **17**: 155–61.

77 PEREZ-IRATXETA, C., P. BORK and M. ANDRADE. 2001. XplorMed: a tool for exploring MEDLINE abstracts. Trends Biochem. Sci. **26**: 573–5.

78 PORTER, M. 1980. An algorithm for suffix stripping. Program. Program. **14**: 130–7.

79 PUSTEJOVSKY, J., J. CASTANO, R. SAURI, A. RUMSHISKY, J. ZHANG and W. LUO. 2002. Medstract: Creating large-scale information servers for biomedical libraries. In Proc. ACL 2002, Philadelphia, PA, USA.

80 RAYCHAUDHURI, S., J. CHANG, P. SUTPHIN and R. ALTMAN. 2002. Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. Genome Res. **12**: 203–14.

81 RAYCHAUDHURI, S. and R. ALTMAN. 2003. A literature-based method for assessing the functional coherence of a gene group. Bioinformatics **19**: 396–401.

82 REBHOLZ-SCHUHMANN, D., H. KIRSCH and F. COUTO. 2005. Facts from text – is text mining ready to deliver? PLoS Biol. **3**: e65.

83 REESE, M., G. HARTZELL, N. HARRIS, U. OHLER, J. ABRIL and S. LEWIS. 2000. Genome annotation assessment in Drosophila melanogaster. Genome Res. **volume?**: 483–501.

84 RZHETSKY, A., T. IOSSIFOV, I. KOIKE, et al. 2004. GeneWays: a system for extracting, analyzing, visualizing, and integrating molecular pathway data. J. Biomed. Inform. **37**: 43–53.

85 SCHULER, G., J. EPSTEIN, H. OHKAWA and J. KANS. 1996. Entrez: molecular biology database and retrieval system. Methods Enzymol. **266**: 141–62.

86 SETTLES, B. 2004. Biomedical named entity recognition using conditional random fields and rich feature sets. In Proc. NLPBA/COLING, Geneva, Switzerland.

87 SHAH, P., L. JENSEN, S. BOUE and P. BORK. 2005. Extraction of transcript diversity from scientific literature. PLoS Comput Biol. **1**: e10.

88 SMITH, L., T. RINDFLESCH and W. WILBUR. 2004. MedPost: a part-of-speech tagger for bioMedical text. Bioinformatics **20**: 2320–1.

89 STAAB, S., C. BLASCHKE, C. NEDELLEC, et al. 2002. Mining information for functional genomics. IEEE Intell. Syst. **17**: 66–80.

90 Stapley, B., L. Kelley and M. Sternberg. 2002. Predicting the subcellular location of proteins from text using support vector machines. Pac. Symp. Biocomput. Hawaii, USA: 374–85.

91 Stoica, E. and M. Hearst. 2006. Predicting gene functions from text using a cross-species approach. Pac. Symp. Biocomput. Hawaii, USA.

92 Tanabe, L. and W. Wilbur. 2002. Tagging gene and protein names in biomedical text. Bioinformatics **18**: 1124–32.

93 Tu, Q., H. Tang and D. Ding. 2004. MedBlast: searching articles related to a biological sequence. Bioinformatics **20**: 75–7.

94 Venclovas, C., A. Zemla, K. Fidelis and J. Moult. 2003. Assessment of progress over the CASP experiments. Proteins. **53**: 585–95.

95 Voorhees, E. and L. Buckland. 2002. The 11th Text REtrieval Conf. (TREC 2002). In Proc. TREC 2002, Gaithersburg, MD, USA.

96 Wattarujeekrit, T., P. Shah and N. Collier. 2004. PASBio: predicate-argument structures for event extraction in molecular biology. BMC Bioinformatics **5**: 155.

97 Weise, C. 2005. eTBlast – It's only words, and words are all I have. Angew. Chem. Int. Ed. Engl. **44**: 182.

98 Wheeler, D., D. Church, S. Federhen, et al. 2003. Database resources of the National Center for Biotechnology. Nucleic Acids Res. **31**: 28–33.

99 Wilbur, W., G. Hazard, G. Divita, J. Mork, A. Aronson and A. Browne. 1999. Analysis of biomedical text for chemical names: a comparison of three methods. Proc. AMIA Symp. Washington, DC, USA: 176–80.

100 Wilbur, W. and L. Coffee. 1994. The effectiveness of document neighboring in search enhancement. Inf. Process Manag. **30**: 253–66.

101 Witten, I., A. Moffat and T. Bell. 1999. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Academic Press, San Diego, CA.

102 Wren, J., W. Hildebrand, S. Chandrasekaran and U. Melcher. 2005. Markov model recognition and classification of DNA/protein sequences within large text databases. Bioinformatics **21**: 4046–53.

103 Xenarios, I., L. Salwinski, X. Duan, P. Higney, S. Kim and D. Eisenberg. 2002. DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Res. **30**: 303–5.

104 Xie, H., A. Wasserman, Z. Levine, A. Novik, V. Grebinskiy and A. Shoshan. 2002. Large-scale protein annotation through gene ontology. Genome Res. **12**: 785–94.

105 Yeh, A., A. Morgan, M. Colosimo and L. Hirschman. 2005. BioCreAtIvE Task 1A: gene mention finding evaluation. BMC Bioinformatics. **6**: S2.

106 Yeh, A., L. Hirschman and A. Morgan. 2003. Evaluation of text data mining for database curation: lessons learned from the KDD Challenge Cup. Bioinformatics. **19**: 331–9.

107 Zhou, G., J. Zhang, J. Su, D. Shen and C. Tan. 2004. Recognizing names in biomedical texts: a machine learning approach. Bioinformatics **20**: 1178–90.

108 Mitchell, A., A. Divoli, J. Kim, M. Hilario, I. Selimas and T. Attwood. 2005. METIS: multiple extraction techniques for informative sentences. Bioinformatics, Vol. 21: 4196–7.

35
Integrating Information for Protein Function Prediction

William Stafford Noble and Asa Ben-Hur

1 Introduction

Most of the work on predicting protein function uses a single source of information – the most common being the amino acid sequence of the protein (see Chapter 30). There are, however, a number of sources of data that are predictive of protein function. These include protein–protein interactions (Chapter 31), the genomic context of a gene (Chapter 32), the protein's structure (Chapter 33), information mined from the biological literature (Chapter 34) and data sources indicating coregulation, such as gene expression and transcription factor binding [10]. A classifier that predicts function based upon several sources should provide more accurate predictions than can be achieved using any single source of data. However, the heterogeneous nature of the data sources makes constructing such a unified classifier challenging.

We have divided the various methods for data integration into five categories (Figure 1). First, *vector-space integration* consists of characterizing proteins in various ways by a set of variables, i.e. as vectors. Any standard classification method can then be applied to the resulting vector-space representation. An alternative – *classifier integration* – is to train a classifier on each source of data and then combine the predictions of the various classifiers into a single prediction. *Kernel methods* are a recent advance in the field of machine learning [41]. These methods provide a coherent framework for integrating various sources of data, applicable even when there is no explicit vector-space representation of the data. Several sources of data form a network that is informative of functional relationships. The prime example of such a network is protein–protein interaction data. Proteins that interact often do so because they participate in the same pathway. Therefore, the network of protein–protein interactions in a cell can be informative of protein function (see Chapter 32). The final two approaches model such networks and their relationship to protein function. Graphical models, both directed and nondirected, provide a probabilistic framework for data integration [23]. Modeling is achieved by representing local probabilistic dependencies; the network

Bioinformatics - From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

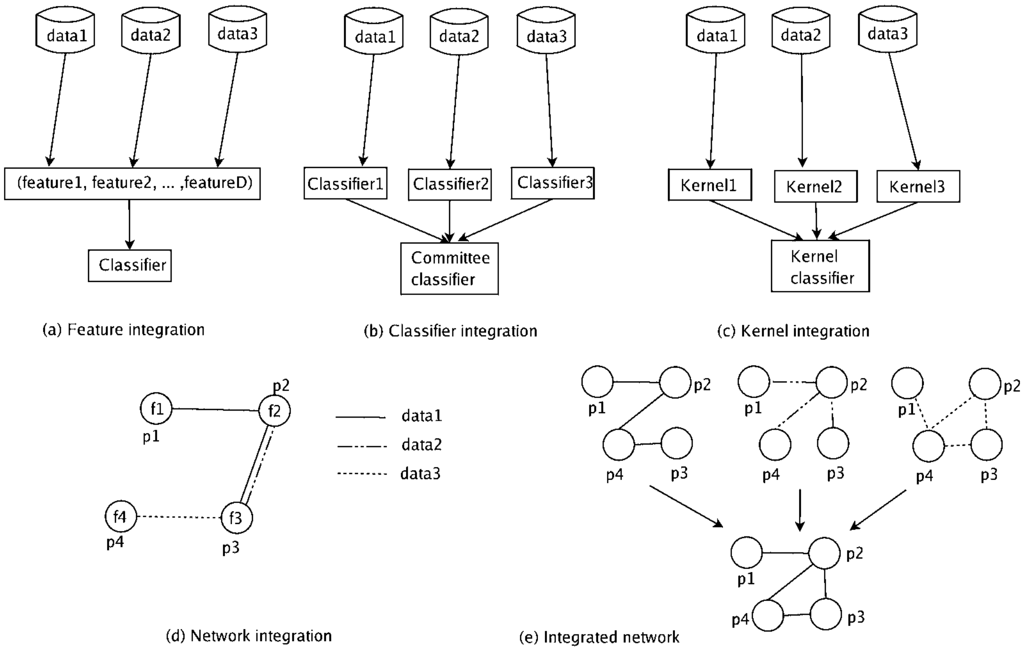


Figure 1 Schematic description of the various methods for integrating genomic information for prediction of protein function. (a) Integration by concatenating data sources into a vector of features. (b) Integrating predictions of several classifiers. (c) Data integration using kernel methods. (d) Integration of network information, typically by Bayesian methods. (e) Integration of several networks of functional relationships into a single network.

structure of these models makes them a natural choice for capturing networks of functional relationships. The last form of integration we discuss does not aim at explicit prediction of protein function, but instead integrates several networks of functional relationships, such as various forms of interaction, coexpression, coregulation, etc., into a single network that unifies all those relationships.

2 Vector-space Integration

Perhaps the simplest form of data integration is to summarize, for each protein, a variety of relevant types of data in a fixed-length vector and feed the resulting collection of vectors into a classification algorithm. This approach has the advantage of simplicity, but treating each type of data identically does not allow us to incorporate much domain knowledge into the design of the classifier. For example, certain sources of data may benefit from a different measure of similarity than others (see Section 4).

An early example of such an approach is described in Ref. [14]. This work presents a limited form of data integration: many different types of protein features are used, but most of these features are derived from the protein's amino acid sequence. Such features include protein length, molecular weight, charge, amino acid composition (i.e. residue frequencies) and isoelectric point. For a subset of the data for which three-dimensional (3-D) structures are available, the authors also include several features based upon secondary structure features; however, the experiments suggest (somewhat surprisingly) that these features are not very informative. The authors apply three different out-of-the-box machine learning algorithms to their data and compare the resulting performance with that of the BLAST sequence comparison algorithm [1] at predicting whether a sequence is an enzyme as well predicting the first two digits of the protein's Enzyme Commission (EC) number. Among the three machine learning algorithms – the C4.5 decision tree, naive Bayes and *k*-nearest-neighbor algorithms – the best-performing algorithm is *k*-nearest-neighbor, which predicts the class of a query protein by finding the most similar protein in the training set and outputting the corresponding label (see Chapters 24 and 27 for more detailed descriptions of this algorithm). This simple approach works as well as BLAST at discriminating between enzymes and nonenzymes, but less well when the task is more specific. The latter result is not surprising, since many of the enzyme classes are characterized by highly specific sequence features [4].

A closely related set of experiments was described 5 years later in Ref. [24]. Like in the previous work, the authors summarize each protein using a fixed-length vector of features derived from the amino acid sequence. After considering 25 such features, the authors settle on 14, which include straightforward features such as average hydrophobicity and number of negatively charged residues, as well as outputs from nine different previously described prediction methods. These predictions include subcellular location, various types of post-translational modifications, low-complexity regions, transmembrane helices, etc. The resulting 14-element feature vectors are given to a feed-forward neural network, which can subsequently predict EC numbers and "cellular role" Gene Ontology (GO) terms with good accuracy.

A larger version of this type of experiment was described in Ref. [46]. In this work, the authors build classifiers for all EC protein families that include 50 or more members (299 families), as well as 233 Pfam families [45]. Each protein is represented using an extremely rich collection of 453 features. These features include statistics derived from the protein sequence, including amino acid frequencies, predicted secondary structure content, molecular weight, average hydrophobicity, isoelectric point, etc. In addition, the authors extract information about sub-cellular location, tissue specificity, etc., from the Swiss-Prot database [3] and encode this information in nine of the 453 features. The paper demonstrates the utility of probabilistic decision trees on this task. This algorithm is essentially an improved version of the C4.5 decision tree classifier, and does a better job of handling unbalanced data sets (when negative examples far outnumber positive examples) and missing data, and which exhibits more stable learning behavior.

Protein structure is often conserved when no significant sequence conservation can be detected. Instead of making predictions using direct structure comparison, one can represent structural features of the protein, analogously to the methods presented earlier, that represent features of the protein sequence [15]. This paper characterizes a protein using several structural features: the total surface area attributable to each residue type, the fractal dimension of the protein surface (which quantifies how "crinkly" the surface of the protein is), surface area to volume ratio, secondary structure content, and the presence of cofactors and metals. Then a support vector machine (SVM) classifier is used to predict the the first digit of the EC number of enzymes whose structure is known. The prediction task prepared by Dobson and Doig [15] is a very difficult one: in each of the six enzyme classes no two structures belong to the same SCOP superfamily. Therefore, sequence-based methods will provide very poor results. The authors have not compared their approach to a sequence-based approach that uses the same set of sequences, so it is unclear whether these structure-based features provide added value. In general, the advantage of using sequence-based classifiers is that many more protein sequences are available than protein structures.

One of the challenges in vector-space integration is determining the contribution of each feature to the accuracy of the classifier and finding small subsets of features that maintain or improve classifier accuracy. This task is known as *feature selection* and is an active area of research in machine learning. The interested reader can find a wealth of information about the state-of-the-art of the field in Refs. [18,19]. The simplest approach to feature selection is the so-called *filter* method whereby one computes for each feature a statistic that reflects how predictive the feature is. Statistics that achieve this goal include the area under the receiver operating characteristic (ROC) curve, the Pearson correlation coefficient, the Fisher criterion score, etc. [18, 19]. Independently scoring each feature does not take into account the redundancy that often exists in high-dimensional data such as gene expression and also ignores the classifier with which the data will be ultimately classified. These issues are handled by *wrapper* or *embedded* methods (see Refs. [18,19]). Wrapper methods use a classifier to evaluate the merit of subsets of features and, as such, can be combined with any classifier. In embedded methods, on the other hand, the classifier is part of the selection process and uses the properties of the classifier to select relevant features. An example of a simple embedded method is the recursive feature elimination (RFE) method [20], that for a linear classifier iteratively removes features for which the magnitude of the corresponding component of the classifer's weight vector is the smallest.

The drawback of vector-space integration is modeling all features the same way. One way to address this issue is to train different classifiers for each source of data and then to combine the predictions of the different classifiers. We call this integration method *classifier integration*, which is described in the next section. *Kernel methods*, which are presented in Section 4, train a single classifier, but allow more flexibility in combining data sources than the vector-space integration methods, by allowing the user to define a different similarity measure for each data source and thereby incorporating more domain knowledge into the design of the classifier. Moreover, kernel methods are applicable in modeling data sources such as protein sequences where no obvious vector-space representation is available.

3 Classifier Integration

The second approach to building a unified protein classification algorithm trains several classifiers and then combines their predictions. Gene finding is a well-known bioinformatics problem for which combining the predictions of several classification methods can provide more accurate predictions [40]. Conceptually, there are several classes of methods for combining the output of different classifiers:

(i) Integration of different classification methods, each trained on the same data.

(ii) Integration of the same method trained on different subsets of the data or on different subsets of features. This is an active field of research in machine learning called *ensemble methods*, and includes methods such as boosting [16], random forests [9] and various "committee machines" [6].

(iii) Integration of several classifiers, each trained on a different source of data.

In our context we are focusing on the third class of methods. The standard method for integrating the results of several classifiers is by a majority vote. A more sophisticated approach is to use a classifier whose job is to integrate the predictions of the various classifiers [6]. Not much work has been done to apply classifier integration methods to protein function prediction. One example of integrating the predictions of several classifiers is described in Refs. [36, 37] and is compared with kernel-based integration in those papers.

See the next section for details. Classifier integration is most useful in cases in which each classifier is available as a black-box, e.g. as is the case for gene finders.

4 Kernel Methods

Recently, a class of algorithms known as kernel methods have become popular in the machine learning community [41, 43] and this popularity has extended into computational biology [42]. A *kernel* is a function that defines similarities between pairs of data objects and a kernel method is an algorithm whose implementation depends on the data only through the kernel. More specifically, a kernel is a similarity measure that satisfies the condition of being a dot product in some space, i.e. $K(x, y)$ can be expressed as $\langle \Phi(x), \Phi(y) \rangle$, where Φ is some possibly nonlinear mapping. This mapping technique has been known for decades, but has gained popularity recently in the context of a particularly powerful classification algorithm, known as the support vector machine (SVM) [8, 12, 50]. The so-called "kernel trick" – mapping data into a higher-dimensional space by means of a predefined kernel function – often results in a problem with more dimensions than examples. The SVM, it turns out, can cope remarkably well with such cases, effectively reducing the curse of dimensionality. Other kernel methods have subsequently been described for classification, regression, clustering, principal components analysis, etc. [43].

Kernel methods provide a coherent framework for data integration. The kernel function provides a form in which to represent a wide variety of data types, including vectors, matrices, strings, trees and graphs. As a kernel method represents data via the kernel function, any data set of n elements can be summarized as an n-by-n matrix of pairwise kernel values. This kernel matrix is a sufficient representation: once it is computed, the original data can be discarded and the kernel method can still perform its function. Furthermore, kernel matrices from different data sources can be combined in a simple kernel algebra, that includes the operations of addition, multiplication and convolution [22]. The simplest way to combine kernels is by adding them: adding kernels is equivalent to concatenating their feature space representations. When the kernels are linear kernels over an explicit vector-space representation this is the same as the vector-space integration described in Section 2. The feature space for the multiplication of kernels is the product of the feature spaces of the kernels. This approach has been used in the context of predicting protein–protein interactions [5].

It is possible to perform vector-space integration with kernel methods. Since kernel methods are sensitive to the scale of each feature, it is often useful to normalize the features such that they are on a similar scale, e.g. by standardizing each feature. When performing integration at the kernel level, an alternative is to normalize the kernel itself, rather than its feature space representation by using a cosine-like kernel $K'(x, y) = K(x, y)/\sqrt{K(x, x)K(y, y)}$, which is the same as projecting the feature-space representation to the unit sphere.

In two related papers, Pavlidis and coworkers apply a kernel-based data integration technique to the problem of protein function prediction [36, 37]. The authors use kernels to combine microarray expression data with phylogenetic profiles and use the resulting combined kernel to train an SVM classifier to place yeast genes into MIPS functional categories [33]. This kernel-based approach is compared to a vector-space integration scheme, which simply concatenates the two types of data into a single vector, and a classifier integration scheme, which trains two different SVMs and then sums the resulting discriminants. In this case, the primary difference between the vector-space integration scheme and the kernel approach is the use of a third-degree polynomial kernel on each data set prior to integration. The polynomial kernel maps each data set into a higher-dimensional space whose features are all monomials over the original features with degree less than or equal to 3. By performing this mapping on each data set individually, rather than on the concatenated vectors, the method incorporates the prior knowledge that inter-feature dependencies within one data set are more likely to be relevant than dependencies between two different types of data. This prior knowledge is borne out by the results, which show that the kernel-based integration scheme provides better classification performance than either of the other two schemes. Data integration by kernel summation has been applied in several other bioinformatics applications: prediction of protein–protein interactions [5] and prediction of metabolic networks [51]. Prediction of metabolic networks, i.e. associating enzymes with metabolic pathways, can be considered a form of function prediction; prediction of pairwise relationships, or networks, is discussed in detail in Section 5.

Rather than simply adding kernels, one can consider a linear combination of kernels, which can take into account how informative each kernel is. For example, if we know that data set A is more useful (i.e. more relevant or less noisy) than data set B, then we can combine the corresponding kernels as a weighted sum: $K_{\mathrm{AB}} = \lambda K_{\mathrm{A}} + K_{\mathrm{B}}$. The only difficulty, of course, is how best to select the data set weighting factor λ. The value of the weighting factor can be set using cross-validation over several choices for its value. This is feasible when combining two kernels. When using a larger number of kernels this is no longer practical and a different approach for weighting the different kernels is required.

Lanckriet and coworkers present a statistical framework for performing kernel-based data integration with weights assigned to each data set [26, 28]. Rather than requiring that the weights be assigned *a priori*, the authors train an SVM and learn the kernel weights simultaneously, using a technique known as semidefinite programming (SDP) [27, 35, 49]. In Ref. [28], this SDP-SVM approach is compared to a previously described Markov random field method for data integration [13] (described in Section 6). Lanckriet and coworkers use the same classification of yeast genes into 13 broad MIPS functional categories and five types of data as [13]: (i) the domain structure of the protein, according to Pfam [45], (ii) known protein–protein interactions, (iii) genetic interactions and (iv) cocomplexed proteins, as identified by the comprehensive yeast genome database, and (v) cell cycle gene expression profiles. Performance is measured using ROC curves [21]. The SDP-SVM approach provides far better performance across all 13 functional classes. A subsequent article [26] applies the same framework to two more yeast classification problems – recognizing membrane proteins and recognizing ribosomal proteins – and provides more details about the SDP-SVM method.

Borgwardt and coworkers propose a kernel method for predicting protein function using protein structure [7]. They represent the structure of a protein as a graph whose nodes are secondary structural elements and whose edges represent proximity in sequence or in 3-D space. The authors propose a kernel that quantifies the similarity between two proteins using the random walk kernel [17], combined with kernels that quantify the similarity between the secondary structural elements of the protein. The proposed kernel thus combines local properties of the protein with the global 3-D structure. They use this kernel with an SVM classifier, as well as with more sophisticated hyper-kernel machinery, to distinguish between enzymes and nonenzymes, and predict the first EC number of an enzyme on a data set used in Ref. [15]. Their more sophisticated approach provides slightly better results than the SVM vector-space integration approach of Dobson and Doig; it is likely that integrating additional structural features into their kernel will provide further improvement.

5 Learning Functional Relationships

Much of the data relevant to predicting the protein function is in the form of a network or can be converted into a network structure. Protein–protein interaction data is an example of such a network: proteins that interact often participate in the same biological process, have a similar localization pattern and, to a lesser extent, have a similar function [5]. Other sources of data that are not directly in the form of a network can be converted into a network structure. Gene expression data can be represented by a graph whose edges represent comembership in a gene expression cluster or weighted by the correlation between the nodes; sequence data can be similarly converted to a graph by means of sequence similarity scores from algorithms such as Smith–Waterman [44] or PSI-BLAST [2]. Other sources of data for weighting edges include similarity of phylogenetic profiles, gene fusion and cocitation from the literature [29] (see also Chapter 32).

Given several networks of pairwise functional relationships, an important task is to unify those networks into a single network [32]. Marcotte and coworkers demonstrate how to combine pairwise functional relationships from three different sources: correlated evolution using phylogenetic profiles [39], correlated mRNA expression profiles and patterns of domain fusion [31]. The data fusion approach is simple: the authors make a list of all pairs of functionally related proteins in yeast according to each method. This list contains 93 000 pairs of proteins. Functional links that are supported by two out of three of the methods are considered "highly confident" and functional annotations from proteins of known function are then propagated across this high-confidence network. This simple approach yielded functional annotations for more than half of the 2557 yeast proteins that were unannotated at the time.

This simple approach of trusting only predictions that are made by more than one method clearly has drawbacks, especially when some of the contributing methods are more reliable than others or when the methods assign confidence values to their predictions. Lee and coworkers address this issue and propose a framework for unifying the scores associated with different networks [29]. The following function assigns a log-likelihood score to a linkage L between two proteins in the presence of a network E in the context of a particular pathway or annotation:

$$LLS(L|E) = \log \frac{P(L|E)/P(\bar{L}|E)}{P(L)/P(\bar{L})},$$

where $P(L|E)$ are the frequencies of the linkage L observed in the data and $\bar{L}$ are instances where the linkage is not observed.

A similar problem is addressed by the MAGIC system [47]; MAGIC estimates the probability that proteins i and j share a functional relationship. The existence of a functional relationship is modeled using several pairwise relationships between proteins: coexpression, colocalization, physical interaction, genetic interactions and comembership in a complex. The paper proposes a Bayesian network model for estimating the probability of a functional relationship. A Bayesian network is a probabilistic model that represents a probability distribution in a form that makes it amenable to efficient computation by encoding the probabilistic dependencies in the data in the form of a directed

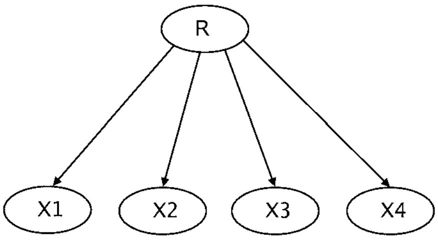


Figure 2 A naive Bayes model: Given that a functional relationship exists between two proteins, the existence of any two relationships between the two proteins (interaction, coexpression, etc.) are independent.

graph (see Refs. [34,38] for textbooks on Bayesian networks). To illustrate the approach we will consider a very simple model. Let R be the random variable that denotes the existence of a functional relationship and let $X_1, \ldots, X_d$ be the pairwise relationships that serve as evidence for the existence of the relationship. We are interested in the probability $P(R|X_1, \ldots, X_d)$. Using Bayes rule, this probability can be expressed as:

$$P(R|X_1, \ldots, X_d) = \frac{P(X_1, \ldots, X_d|R)P(R)}{P(X_1, \ldots, X_d)}.$$

The naive Bayes model is the assumption that each data source X_i is conditionally independent of the other data sources, i.e. $P(X_i|R, X_j) = P(X_i|R)$ for $j \neq i$ [34,38]. This assumption enables us to write:

$$\begin{aligned}
P(X_1, \ldots, X_d|R) &= P(X_1|R)P(X_2, \ldots, X_d|X_1, R) = P(X_1|R)P(X_2, \ldots, X_d|R) \\
&= P(X_1|R)P(X_2|R)P(X_3, \ldots, X_d|X_2, R) \\
&= P(X_1|R)P(X_2|R)P(X_3, \ldots, X_d|R) \\
\ldots &= \prod_{i=1}^{d} P(X_i|R) \, .
\end{aligned}$$

Finally, we have:

$$P(R|X_1, \ldots, X_d) = \frac{\prod_i P(X_i|R)P(R)}{\prod_i P(X_i)}. \tag{1}$$

The classifier resulting from this independence assumption is known as *naive Bayes*. The independence assumption underlying the naive Bayes classifier can be expressed as the directed graph shown in Figure 2. The interpretation of the network structure is that a particular pairwise relationship between two genes, e.g. physical interactions, is a consequence of the existence of a functional relationship between the genes. The Bayesian network suggested by Troyanskaya and coworkers introduces some dependencies between the various sources of data, but the general dependency structure is similar to the one presented here. The conditional probability distributions at the nodes of the network were determined by surveying a panel of experts in yeast molecular biology. Thus, there is no learning involved in the construction of the system. The pairwise relationships used in MAGIC are matrices whose i, j element is the degree to which protein i and j share a particular relationship. For some data sources this is a binary score, e.g. the proteins coded physically interact or their genes belong to the same gene expression cluster. For other data sources, the score is continuous, e.g. when measuring expression correlation.

Learning of metabolic networks is an example of learning of functional relationships. In this problem, one learns a network whose nodes are enzymes and whose edges indicate that the two enzymes catalyze successive reactions in a pathway. Yamanishi and coworkers [51] integrate many sources of data in the context of kernel methods and consider two approaches to this problem. (i) A "direct" approach: a classifier is trained on positive examples – pairs of enzymes that are known to belong to the metabolic network versus pairs of enzymes that are not part of the network. (ii) "Learning the feature space": before training a classifier, a low-dimensional feature space is computed. This space captures the proximity between enzymes that belong to the metabolic network. Yamanishi and coworkers find that learning the feature space significantly improves the results, and, further, that integration of several kernels based on expression data, phylogenetic profiles, localization and chemical compatibility gives better results than any individual kernel.

6 Learning Function from Networks of Pairwise Relationships

When pairwise relationships between proteins are known, the function of unknown proteins can be inferred using the "guilt by association" rule by looking at the annotations of its neighbors in the network. This rule assigns a functional annotation using a majority vote among the annotations of the neighboring nodes. This method of assigning function is clearly an oversimplification of the problem since it ignores the larger context in which a node appears. An alternative approach is to integrate information across the network, rather than relying only upon local information. In this section, we describe several approaches that consider the network as a whole when making predictions. All the methods predict a single function of interest using a network in which each protein is a node. Binary node labels indicate whether the protein has the function of interest and a third label value can be added to represent proteins with unknown function. The nodes are connected with edges whose weights reflect the degree to which the two proteins are related. Multiple networks can be used either by merging the networks or by having several networks sharing the annotation variables.

Deng and coworkers proposed a Markov random field (MRF) model to take into account multiple networks of relationships between genes [13]. Relationships such as protein–protein interaction and coexpression are symmetric: no directionality can be assigned to such relationships. Therefore, Bayesian networks that rely on a directed graph to model the dependencies between variables are not readily applicable. MRFs, which represent probabilistic dependencies using undirected graphical models, are therefore a more appropriate modeling choice. The reader is referred to Ref. [38], for example, for an in-depth discussion of MRFs. Deng and coworkers estimate an MRF model for each function of interest, and each protein is assigned a variable X_i with a state of either 1 or 0, depending on whether or not the protein has that function. The joint probability of X, the vector of variables, is written as: $\exp(-U(x))/Z(\theta)$, where x is a value of X, $Z(\theta)$ is a normalization factor that depends on the parameters of the model and:

$$\begin{aligned} U(x) \quad = \quad & -\alpha \sum_{i=1}^{N} x_i - \beta \sum_{(i,j)\in S} \left[(1-x_i)x_j + x_i(1-x_j)\right] \\ & - \gamma \sum_{(i,j)\in S} x_i x_j - \kappa \sum_{(i,j)\in S} (1-x_i)(1-x_j)\,, \end{aligned} \tag{2}$$

where S is the set of edges of the graph, $\alpha = \log(\frac{\pi}{1-\pi})$ and π is the prior probability for observing the function of interest. The first term in Eq. (2) represents the prior probability of observing the configuration x. The rest of the terms represent interactions between neighbors in the network: the first counting the number of neighbors that do not agree on the assignment of function, the second counting the neighbors that share the function of interest and the third counting neighbors that are negative examples.

The training data is a subset of the proteins whose function is known. Using this data, the probability distribution of the function of the rest of the proteins is estimated by conditioning on the state of the known proteins. The probability of an unknown protein having the function of interest can then be obtained by summing over the possible configurations of the rest of the unknown proteins. The authors propose a Gibbs sampling scheme for estimating these probability distributions.

So far we presented the MRF model that uses a single network. When several networks are available, the probability distribution is a product of terms, each of which is of the form (2), sharing the same values of X. The authors also add a component that takes into account the domain composition of the given proteins (see Ref. [13] for the details). The prior probabilities for a protein to be assigned the function of interest is determined using data on protein complexes, according to the fraction of members of the complex that have that function. When that information is not available, a global prior based on the frequency of the annotation is used. Given a set of possible annotations, one can estimate the probabilities of each annotation. Multiple annotations can then be assigned on the basis of the assigned probabilities. The correlations between different annotations are not taken into account.

The authors apply their method to classify yeast proteins using MIPS annotations combining networks from several sources:

- Physical interactions taken from the MIPS database, including 2448 interactions between 1877 proteins.
- MIPS genetic interactions.
- Comembership in a complex obtained using TAP data, including 232 complexes involving 1088 proteins with known function.
- Cell cycle gene expression data. A network is formed by forming edges between proteins whose expression is above some threshold (0.8 in the paper).

The authors find that combining multiple sources of data improves their performance relative to learning from any single data source.

Karaoz and coworkers propose a general framework for integrating and propagating evidence in functional linkage networks [25]. This approach is a generalization of the "guilt by association" rule which, in essence, repeatedly applies the rule until the network reaches a state that is maximally consistent with the observed data. They begin with a functional linkage network in which the edges are defined by protein–protein interactions and the edge weights are defined by correlating the corresponding mRNA expression profiles. A separate network is defined for each functional annotation (GO term) and each node in the network is assigned a label based upon whether the protein is assigned the current GO term (1), a different GO term in the same GO hierarchy (-1) or no GO term at all (0). The optimization procedure attempts to assign labels (1 or -1) to the zero-labeled nodes so as to maximize an "energy" function. Their energy function is similar to the one used in Ref. [13], but the approach is limited to a single network and, rather than assigning function according to the distribution of a variable, they use a local minimum of the energy function. As described, the method integrates two types of data: one used in the definition of the network topology and the other defines the edge weights. A larger number of sources of data can be integrated by performing a preprocessing step of network integration by one of the methods described in the previous section so that the network or the weighting are computed by more than a single source of data. A similar approach is described in Ref. [11]; to address the situation that a node has no annotated neighbors and "guilt by association" cannot be applied at the node, their model has a state space that indicates whether "guilt by association" is ready to be applied at a node.

Rather than trying to estimate a probabilistic network model, one can take a more direct approach of making predictions on the protein graph. An example of this approach is found in Ref. [48]. This paper follows the standard machine learning paradigm of optimizing a two-part loss (or fitness) function. This function is composed of an error term that measures how well the predicted function follows the training data (existing annotations) and a regularization term that ensures the "smoothness" (regularization, in machine learning terms) of the predicted biological function. In the context of learning on a graph, smoothness means that adjacent nodes have similar predicted function. As in the previous approaches, the authors define a graph whose nodes are proteins labeled as "+1" or "−1", depending on whether the protein is annotated with the function of interest. Let n be the number of proteins, and assume that the function of the first p proteins is known and is given by a vector y with elements equal to ±1. Proteins with unknown function have $y_i = 0$. They define a variable f_i which is the predicted annotation of node i. The value of f is estimated by minimizing the function:

$$\sum_{i=1}^{p}(f_i - y_i)^2 + \mu \sum_{p+1}^{n} f_i^2 + \sum_{i,j} w_{ij}(f_i - f_j)^2 \,,$$

where w_{ij} is the weight on the edge connecting nodes i and j. The first term is the error term; the rest are regularization terms: the second term ensures that the value of f for unlabeled nodes is bounded and the third term ensures that adjacent nodes have a similar annotation. Setting $\mu = 1$, this expression can be written as:

$$\sum_{i=1}^{n}(f_i - y_i)^2 + \sum_{i,j} w_{ij}(f_i - f_j)^2$$

where we take into account that the last $n - p$ entries of y are zero. In order to facilitate extending this formulation to multiple networks this is written as:

$$\min_{f,\gamma} \sum_{i=1}^{n}(f_i - y_i)^2 + c\gamma, \;\; f^{\mathrm{T}} L f \leq \gamma \,,$$

where L is the Laplacian matrix $L = D - W$, where $D = \mathrm{diag}(d_i)$, $d_i = \sum_j w_{ij}$. Multiple networks are incorporated as:

$$\min_{f,\gamma} \sum_{i=1}^{n}(f_i - y_i)^2 + c\gamma, \;\; f^{\mathrm{T}} L_k f \leq \gamma,$$

where L_k is the Laplacian for network k. Sparsity and a dual formulation of the problem yield efficient algorithms that enable solving the problem even for large networks. The authors apply their method to predict MIPS categories in yeast (the same data used in Refs. [13, 28]). They use networks based on Pfam domain composition similarity, coparticipation in a protein complex, MIPS physical interactions, genetic interactions and cell cycle gene expression similarity. They obtain similar performance to that of the SDP-SVM method [28] and better than the MRF method [13].

7 Discussion

The methods described in this chapter illustrate that integrating several sources of data provides improved accuracy in protein function prediction. We described several approaches for data integration. Selecting among these various approaches is difficult, because a large-scale experimental comparison of different integration techniques has not been performed. In the end, researchers tend to select the modeling technique that they are most comfortable with.

A related problem for which data integration yields improved classification accuracy is prediction of protein–protein interactions and the related problem of prediction of comembership in a complex. In this domain, examples of vector-space integration include the works described in Refs. [30, 52]. These papers use a collection of features to predict comembership in a complex using probabilistic decision trees and random forests. In these experiments, features include microarray experiment correlations, transcription factor-binding data, localization, phenotype, gene fusion, gene neighborhood and phylogenetic profiles. An example of kernel-based integration is found in Ref. [5] where sequence-based kernels are used in conjunction with kernels based on features such as GO annotations to predict protein–protein interactions. The use of kernel-based classifiers allows the use of a high-dimensional feature-space that cannot be represented explicitly in practice. As in the other examples presented in this chapter, the combined method performs significantly better than methods that use only a single source of data.

References

1 S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman.. A basic local alignment search tool. J. Mol. Biol. **215**: 403–10.

2 S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. Nucleic Acids Res. **25**: 3389–3402, 1997.

3 A. Bairoch. The SWISS-PROT protein sequence data bank: Current status. Nucleic Acids Res. **22**(17): 3578–80, 1994.

4 A. Ben-Hur and D. Brutlag. Protein sequence motifs: Highly predictive features of protein function. In:

I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, (eds.), *Feature extraction, foundations and applications*. Springer, Berlin, in press.

5 A. Ben-Hur and W. S. Noble. Kernel methods for predicting protein-protein interactions. Bioinformatics **21** (Suppl. 1): i38–46, 2005.

6 C. Bishop. Pattern Recognition and Machine Learning. Springer, 2006.

7 K. M. Borgwardt, C. S. Ong, S. Schoenauer, S. V. N. Vishwanathan, A. Smola, and H.-P. Kriegel. Protein function prediction via graph kernels. Bioinformatics **21**(Suppl. 1): i47–56, 2005.

8 B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In: Proc. 5th Ann. ACM Workshop on COLT, 144–152, Pittsburgh, PA, 1992.

9 L. Breiman. Random forests. Machine Learning **45**(1): 5–32, 2001.

10 M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, Jr. M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector machines. Proc. Natl Acad. Sci. USA **97**: 262–7, 2000.

11 Y. Chen and D. Xu. Genome-scale protein function prediction in yeast *Saccharomyces cerevisiae* through integrating multiple sources of high-throughput data. Pac. Symp. Biocomput. **10**: 471–82. 2005.

12 N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, 2000.

13 M. Deng, T. Chen, and F. Sun. An integrated probabilistic model for functional prediction of proteins. Proc. RECOMB **7**: 95–103, 2003.

14 M. des Jardins, P.D. Karp, M. Krummenacker, T.J. Lee, and C.A. Ouzounis. Prediction of enzyme classification from protein sequence without the use of sequence similarity. Proc. ISMB **5**: 92–9, 1997.

15 P. D. Dobson and A. J. Doig. Predicting enzyme class from protein structure without alignments. J. Mol. Biol. **345**: 187–99, 2005.

16 Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55(1)**: 119–39, 1997.

17 T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In: Proc. 16th Annu. Conf. on Computational Learning Theory and the 7th Annu. Workshop on Kernel Machines, Washington, DC, USA: 129–43. 2003.

18 I. Guyon. Special issue on variable and feature selection. J. Machine Learn. Res. **3** (March) 2003.

19 I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh (eds.) *Feature Extraction, Foundations and Applications*. Springer, Berlin, 2006.

20 I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. Machine Learn. **46**: 389–422, 2002.

21 J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristic (ROC) curve. Radiology **143**: 29–36, 1982.

22 D. Haussler. Convolution kernels on discrete structures. *Technical Report UCSC-CRL-99-10*, University of California, Santa Cruz, CA, 1999.

23 F. V. Jensen. *Bayesian Networks and Decision Graphs*. Springer, Berlin, 2001.

24 L. J. Jensen, R. Gupta, N. Blom, et al. Prediction of human protein function from post-translational modifications and localization features. J. Mol. Biol. **319**: 1257–65, 2002.

25 U. Karaoz, T. M. Murali, S. Letovsky, Y. Zheng, C. Ding, C. R. Cantor, and S. Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. Proc. Natl Acad. Sci. USA **101**: 2888–93, 2004.

26 G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. Bioinformatics **20**: 2626–35, 2004.

27 G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan. Learning the kernel matrix with semi-definite programming. In: Proc. 19th Int. Conf. on Machine Learning, Sydney: xxx–xxx, 2002.

28 G. R. G. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based data fusion and its application to protein function prediction in yeast. Pac. Symp. Biocomput.: 300–11, 2004.

29 I. Lee, S. V. Date, A. T. Adai, and E. M. Marcotte. A probabilistic functional network of yeast genes. Science **306**: 1555–8, 2004.

30 N. Lin, B. Wu, R. Jansen, M. Gerstein, and H. Zhao. Information assessment on predicting protein-protein interactions. BMC Bioinformatics **5**: 154, 2004.

31 E. M. Marcotte, M. Pellegrini, H.-L. Ng, D. W. Rice, T. O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. Science **285**: 751–3, 1999.

32 E. M. Marcotte, M. Pellegrini, M. J. Thompson, T. O. Yeates, and D. Eisenberg. A combined algorithm for genome-wide prediction of protein function. Nature **402**: 83–6, 1999.

33 H. W. Mewes, D. Frishman, C. Gruber, et al. MIPS: a database for genomes and protein sequences. Nucleic Acids Res. **28**: 37–40, 2000.

34 R. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, Englewood Cliffs, NJ, 2003.

35 Y. Nesterov and A. Nemirovsky. *Interior-point Polynomial Methods in Convex Programming: Theory and Applications*, volume 13 of *Studies in Applied Mathematics*. SIAM, Philadelphia, PA, 1994.

36 P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy. Gene functional classification from heterogeneous data. In: Proc. 5th Annu. Int. Conf. on Computational Molecular Biology, Montreal, Canada: 242–8, 2001.

37 P. Pavlidis, J. Weston, J. Cai, and W. S. Noble. Learning gene functional classifications from multiple data types. J. Comput. Biol. **9**: 401–11, 2002.

38 J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1998.

39 M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. Proc. Natl Acad. Sci. USA **96**: 4285–8, 1999.

40 S. Rogic, B. F. Ouellette, and A. K. Mackworth. Improving gene recognition accuracy by combining predictions from two gene-finding programs. Bioinformatics **18**: 1034–45, 2002.

41 B. Schölkopf and A. Smola. *Learning with Kernels*. MIT Press, Cambridge, MA, 2002.

42 B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. Proceedings of the 7th International Conference on Artificial Neural Networks **1327**: 583, 1997.

43 J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, Cambridge, 2004.

44 H. O. Smith, T. M. Annau, and S. Chandrasegaran. Finding sequence motifs in groups of functionally related proteins. Proc. Natl Acad. Sci. USA **87**: 826–30, 1990.

45 E. Sonnhammer, S. Eddy, and R. Durbin. Pfam: a comprehensive database of protein domain families based on seed alignments. Proteins **28**: 405–20, 1997.

46 U. Syed and G. Yona. Using a mixture of probabilistic decision trees for direct prediction of protein function. *Proc. RECOMB*: 289–300, 2003.

47 O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein. A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *S. cerevisiae*). Proc. Natl Acad. Sci. USA **100**: 8348–53, 2003.

48 K. Tsuda, H. J. Shin, and B. Schölkopf. Fast protein classification with multiple networks. Bioinformatics **21**: ii59–65, 2005.

49 L. Vandenberghe and S. Boyd. Semidefinite programming. SIAM Rev. **38**: 49–95, 1996.

50 V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.

51 Y. Yamanishi, J.-P. Vert, and M. Kanehisa. Supervised enzyme network inference from the integration of genomic data and chemical information. Bioinformatics **21**: i468–77, 2005.

52 L. V. Zhang, S. L. Wong, O. D. King, and F. P. Roth. Predicting co-complexed protein pairs using genomic and proteomic data integration. BMC Bioinformatics **5**: 38–53, 2004.

36
The Molecular Basis of Predicting Druggability

Bissan Al-Lazikani, Anna Gaulton, Gaia Paolini, Jerry Lanfear, John Overington, and Andrew Hopkins

1 Introduction

Medicinal chemists have learnt through the experience of many hundreds of screening campaigns in the pharmaceutical industry that for many targets no small-molecule modulators have yet been discovered, even when screened against a diverse chemical file of hundreds of thousands to millions of compounds. Even when the medicinal chemist is fortunate enough to discover a small-molecule modulator of the biological target of interest, it is common for many 'leads' compounds to be unsuitable for optimization into prototype drugs. Chemical biologists may not require such optimized chemical tools, but both the chemical biologist and the medicinal chemist can learn from each other in their experience of discovering chemical tools and leads. The failure of many screening campaigns to discover drug-like leads or chemical tools against certain targets has lead to two competing hypotheses to explain and overcome this phenomenon. The first hypothesis is that the discovery of a chemical tool against a target is a function of the diversity of chemical space screen against the target, independent of the target – the *diversity argument*. The second hypothesis claims that the ability to discover a small-molecule modulator is an inherent property of the physicochemical topology of a biological target, independent of chemical space – the *druggability argument*. These constraints are more severe if the aim is to discover drugs that can be orally administered. The concept of *druggability* postulates that since the binding sites on biological molecules are complementary in terms of volume, topology and physicochemical properties to their ligands, then only certain binding sites on putative drug targets are compatible with binding compounds with high affinity to compounds with "drug-like" properties [15]. Furthermore, the concept also asserts that molecular recognition on biological targets, such as proteins, has evolved to be exquisitely specific at discrete sites on protein surfaces and creates stringent physicochemical limits that restrict the target set available to modulation by small molecules. The extension of

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

this concept to a whole genome analysis leads to the identification of the *druggable genome* – the genes and their expressed proteome predicted to be amenable to modulation by compounds compatible with drug-like properties [14,25].

2 Chemical Properties of Drugs, Leads and Tools

For *in vitro* or cellular experiments the chemical biologists would require a minimum set of physicochemical characteristics of the compound to ensure that the compound is within a range of solubility and polar/hydrophobic balance of properties that enable the tool to permeate the cell membrane and reach the site of action. For the medicinal chemist, the same principles apply, but the great range of biological barriers that a drug needs to pass through in order to affect the biological system of a whole organism is far greater and thus reduces the molecular property range of chemical space. Lipinski introduced the concept of physicochemical property limits to the drugs, with respect to solubility and permeability of drugs from a seminal analysis of the Derwent World Drug Index which demonstrated orally administered drugs are far more likely to reside in areas of chemical space defined by a limited range of molecular properties. Lipinski's analysis demonstrated that 90% of orally absorbed drugs had molecular weights of less than 500 Da, less than five hydrogen-bond donors (such as the OH and NH group count), fewer than 10 hydrogen-bond acceptors (such as the total, combined nitrogen and oxygen atom count being 10 or less) and lipophilicity less than calculated $\log P \leq 5$ [20]. The multiples of five observed in the molecular properties of drugs led to the coining of the term *Lipinski's "Rule of Five"*. Since the work of Lipinski and coworkers, various expansions of the definition and methods to predict "drug-likeness" have been proposed in the literature [1,9,17,19–22, 27,28,31,34–36]. The common thread emerging from the field is that drug-likeness is defined by a range of molecular properties and descriptors that can discriminate between drugs and nondrugs for such characteristics as oral absorption, aqueous solubility and permeability. This is illustrated by the observation that the distribution of mean molecular properties of approved oral (small-molecule) drugs has changed little in the past 20 years, despite changes in the range of indications and targets [33].

3 Molecular Recognition is the Basis for Druggability

The molecular basis of the *a priori* druggability hypothesis derives from biophysical study of molecular recognition. The binding energy (ΔG) of a ligand to a molecular target (e.g. protein, RNA, DNA, carbohydrate) is defined as:

$$\Delta G = -RT \ln K_i = 1.4 \log K_i \tag{1}$$

where R is the gas constant (1.986 cal mol^{-1} K^{-1}).

The affinity of binding is predominately driven by the van der Waals and entropy components of the binding energy by the burying of hydrophobic surfaces. Thus for a ligand, such as a drug molecule, to bind with an affinity of K_i = 10 nM it requires a binding energy (ΔG) of –11 kcal mol^{-1}. A lower affinity "hit" from a high-throughput screen of K_i = 1 μM affinity equates to 8.4 kcal mol^{-1}. Thus, a 10-fold increase in potency is equivalent to 1.36 kcal mol^{-1} of binding energy. The binding energy potential of a ligand is, in general, proportional to the available surface area and its properties. The hydrophobic effect from the displacement of water and the van der Waals attractions between atoms contributes approximately 0.03 kcal mol^{-1} $\mathrm{Å}^{-2}$. Thus a ligand with a 10 nM dissociation constant would be required to bury 370 $\mathrm{Å}^2$ of hydrophobic surface area, assuming there are no strong ionic interactions between the protein and the ligand. Empirical analysis of nearly 50 000 biologically active drug-like molecules reveals a linear coloration between molecular weight and molecular surface area (Figure 1). The contribution of the hydrophobic surface to binding energy is demonstrated by the phenomenon of the "magic methyl", where experienced medicinal chemists often observe that a single methyl group, judiciously placed, can increase ligand affinity by 10-fold, approximately equivalent to the maximal affinity per nonhydrogen atom [16]. The accessible hydrophobic surface area of a methyl group is approximately 46 $\mathrm{Å}^2$(if one assumes all of the hydrophobic surface area is encapsulated by the protein binding site and thus forms full contact with the target) with a hydrophobic effect of 0.03 kcal mol^{-1} $\mathrm{Å}^{-2}$ equal approximately to 1.36 kcal mol^{-1}, equivalent to the observed 10-fold affinity increase. In addition to the predominantly hydrophobic contribution to the binding of many drugs, ionic interactions, such as those found in zinc proteases (such as ACE inhibitors) contribute to the binding energy. The attraction of complementary polar groups contributes up to up to 0.1 kcal mol^{-1} $\mathrm{Å}^{-2}$, with ionic salt bridge approximately 3 times greater, enabling low-molecular-weight compounds to bind strongly. Unlike hydrophobic interactions, complementary polar interactions are dependent on the correct geometry. Thus, encapsulated cavities are capable of binding low-molecular-weight compounds with high affinities since they maximize the ratio of the surface area to the volume.

Thus, the physicochemical characteristics of the binding site define the physical and chemical properties of the ligand. Therefore, a target needs a pocket, whether the pocket is predefined or formed on binding by allosteric mechanisms. In general, thermodynamics and selection pressure play a part in reducing the accidental existence of such favorable pockets for ligand

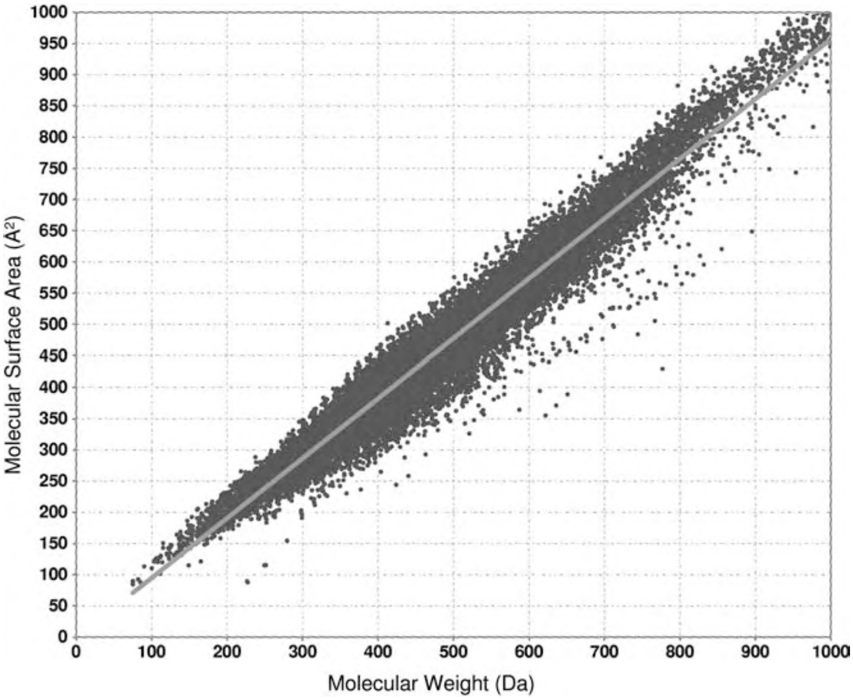


Figure 1 Relationship between molecular weight and molecular surface area. Analysis of 49 456 biologically active, drug-like compounds with $IC_{50} \leq 100$ nM. Molecular weight was calculated from the chemical structures represented as desalted, canonical SMILES strings. The calculated molecular surface area of N, O, P and S atoms was estimated using the fast Ertl method [10] using a two-dimensional approximation. All other atom types (excluding H atoms) were estimated using an overlapping spheres method. All calculations were performed using Scitegic's Pipeline Pilot (San Diego, CA.).

interactions. The thermodynamic argument contests that it costs energy to maintain an exposed hydrophobic pocket in an aqueous environment. Selection pressure may also increase the specificity of molecular recognition for ligand pockets to avoid inappropriate signaling or inactivation from the milieu of metabolite and small molecules in which cells are bathed.

A quantitative approach is already well established for assessing the drug-like properties of a small molecule, could such a quantitative approach for assessing the properties of proteins as drugs? The "Rule of Five" is a set of properties to suggest which compounds are likely to show poor absorption or permeation, since such compounds are unlikely to show good oral bioavailability [20]. Physicochemical constraints such as this limit the type of proteins we see as drug targets; simply put, drug targets need to be able to bind compounds with complementary properties. As a receptor binding site must be complementary to a drug, it is reasonable to assume that equivalent rules could be developed to describe physicochemical properties of binding sites with the potential to bind "Rule of Five" compliant molecules with a potent binding constant (e.g. $K_i < 100$ nM). A number of properties complementary to the "Rule of Five" can be calculated, e.g. the surface area and volume of the pocket, hydrophobic and hydrophilic character, and the curvature and shape of the pocket. Following the assumption that properties of the drug are complementary to those of the binding site, analysis of the calculated physicochemical properties of the putative drug binding pocket on the target protein can provide an important guide to the medicinal chemist in predicting the likelihood of discovering a drug against the particular target site. Based on the known physicochemical properties of passively absorbed oral drugs, one would predict "druggable" binding sites to be predominately apolar cavities of 400–1000 Å^{3}'where over 65% of the pocket is buried or encapsulated, with an accessible hydrophobic surface area of at least 350 Å^{2}.

Druggability predictions have been empirically explored using heteronuclear nuclear magnetic resonance (NMR) to identify and characterize the binding surfaces on protein by screening around 10 000 low-molecular-molecules (average molecular weight 220, average cLog$P$ 1.5) [13]. Screening results from 23 proteins reveal 90% of the ligands binds to sites know to be a small molecule ligand binding sites. In the relative small sample of proteins studied, Hajduk and coworkers note a high correlation between experimental NMR hit rates and the ability to find a high affinity ligands. Only in three out of the 23 proteins were distinct uncompetitive new binding sites were discovered. The authors' postulated that these new sites could possibly play an unknown physiological role in the proteins functions.

4 Estimating the Size of the Druggable Genome

Whilst our current knowledge may be limited in predicting *a priori* where uncompetitive allosteric binding sites may appear from a protein sequence, we may be able to identify at the sequence and structural level which targets are more likely to be potentially amenable to modulation by drug-like small molecules from extrapolation of our current knowledge.

Using knowledge of proteins to which current drugs and leads bind, we can infer the subset of the human genes and protein that have a high probability of being potentially druggable, i.e. capable of binding drug-like small molecules with high affinity. Outlined below are a number of methodologies and approaches that have been used to infer the druggable portion of targets encoded by the human genome. In this chapter we have extended the work of Hopkins and Groom, and attempted to estimate the size of the druggable human genome using three distinct methodologies:

- Homology-based analysis from comprehensive survey of drugs and leads
- Feature-based probabilistic druggability analysis
- Structure-based amenability analysis

4.1 Initial Estimates

In order to begin to gauge the number of possible drug targets in the human genome, one should begin with a survey of the knowledge of the current modes of action of existing drugs. In a review of the pharmacological literature, Drews [7, 8] identified 483 targets for known drugs. From this figure Drews later estimated the number of ligand binding domains as a measure of the number of potential points at which small molecule therapeutic agents could be close to 10 000; however, the methodology of how these numbers were derived is not disclosed [6].

4.2 Hopkins and Groom's Method

The first systematic survey of the druggable genome, following the publication of the draft human genome [18, 32], was by Hopkins and Groom [14]. Hopkins and Groom attempted to identify the genes which produced potentially druggable proteins by their membership of druggable gene families. The explicit assumption of a gene family-based analysis is that the conserved architecture of the druggable protein domain is likely to be conserved amongst related members of that domain's gene family. Hopkins and Groom approached the problem in two stages. (i) A database of drug target sequences from a comprehensive survey the literature and investigation drug databases was complied. (ii) The constructed drug target sequence database as used to identify related members of a putative druggable gene family from the protein domain annotation of the translated human protein sequences. Hopkins and Groom's analysis of the literature, the Investigational Drugs Database and the PharmaProjects databaseidentifies 399 nonredundant molecular targets shown to bind "Rule of Five" compliant compounds, with binding affinities below 10 μM. Whilst there is some degree of overlap with Drews' work [7, 8], a significant amount of redundancy was observed in the initial study. In addition a number of new proteins targeted by experimental drugs were captured. Likewise, some targets for biological agents, for which modulation by "Rule of Five" compliant compounds has not yet been shown, were eliminated from the survey. Nearly half of the targets fall into just six major gene families: G-protein-coupled receptors (GPCRs), serine/threonine and tyrosine protein kinases superfamily, zinc metallopeptidases, serine proteases, nuclear hormone receptors, and phosphodiesterases (PDEs). Of the 399 targets of marketed and experimental drugs identified, 376 sequences could be assigned to 130 drug-binding domains, as captured by their InterPro domain annotation. Of these, 125 are domains with homologs and orthologs present in the human proteome. The sequence and functional similarities within a gene family assume a general conservation of binding site architecture between family members. The explicit assumption being that if one member of a gene family is modulated by a drug molecule, other members of the family could also be able to bind a compound with similar physicochemical properties. Following the above logic, 3051 genes were identified as belonging to the 125 druggable InterPro domains and thus predicted to encoded proteins that have some precedence for inferring their ability to bind a drug-like molecules.

Hopkins and Groom's database identifies only 120 biological targets as the modes of action for marketed, "Rule of Five" compliant drugs – significantly less than the previous estimate that launched drugs acted on 483 targets. Interestingly, the vast majority of the drugs and leads identified in this survey, about 90%, are competitive with an endogenous ligands at a structurally defined binding site. This figure is similar to the rates of discovering new binding sites by Hajduk and coworkers [13] (Hopkins, personal communication).

4.3 Orth and Coworkers Update 2004

Orth and coworkers [24] based an estimate on the druggable gene families on the InterPro domain assignments in the annotated gene-encoding loci of the 2004 release of the Consensus CoDing Sequence (CCDS) database. The authors estimate the 3080 nonredundant gene-encoded loci in the human genome predicted to be belonging to the druggable genome with over 2950 druggable gene sequences in public databases.

4.4 Russ and Lampel's Update 2005

Russ and Lampel [30] conducted an estimate on the druggable genome based on the preliminary final assembly (Ensemble Release 35) of the human genome where 99% of the sequence has high quality cover. The authors found Pfam protein domain annotation predicted fewer false positives than the InterPro classification used by Hopkins and Groom [14], estimating 3100 druggable genes from the previously defined set of druggable protein domains, approximately 2900 of which were predicted by both approaches. Of the 3100 predicted genes, 2600 are covered by the consensus CCDS annotation of the major genome databases. Extrapolation from the manual VEGA genome annotation databases (about 40% of total genome) leads the authors to a conservative estimate of around 2500 druggable genes. The authors consider these assessments from the high confident gene prediction databases to be a considered lower conservative estimate of the size of the druggable genome.

5 Homology-based Analysis of Drug Targets

In order to expand the homology analysis methodology for identifying which targets expressed from the human genome are likely to be druggable, it is necessary to expand our survey to identifying all known biological targets of drugs and lead compounds. Inpharmatica commissioned the construction of two databases, DrugStore and StARLITe, to accurately ascertain the number of biological targets modulated by drugs and preclinical medicinal chemistry compounds, respectively.

Inpharmatica's DrugStore is a relational database relating all Food and Drug Administration approved drugs to their molecular targets and approved indication. From this analysis we have identified 26 000 drugs products which reduces to 1783 unique new molecular entities (NMEs), of which 1415 are small molecule chemical entities, 180 are biological therapeutics (18 of which are antibodies), and the remainder are vitamins and supplements. As drug discovery has been more target-centric over the past two decades in its research *modus operandi*, a key point of debate has been how many modes of action acted upon by approved drugs? The first attempt to ascertain this number was by Drews, who estimated known drugs acted on 483 targets – the source of the often quoted "500 targets" figures. Hopkins and Groom's analysis challenged this figure and suggested, irrespective of poly-pharmacology off-target effects, "Rule of Five" compliant (orally administered) approved drugs acted primarily on only 120 modes of action. A subsequent analysis by Burgess and Golden proposed all approved NMEs consisting of new chemical entities (NCEs) and new biological entities (NBEs) targeted 272 proteins [3, 5, 11, 12]. Here, we propose from analysis of the DrugStore database all NMEs primarily act on 301 drugs targets of which 238 are human proteins and only 170 are human proteins targeted by small-molecule drugs (Table 1 and Figure 2). Biological drugs target 59 modes of action with current marketed antibody therapeutics acting on 15 human targets. Only nine targets are currently found to be modulated by both small-molecule and biological drugs. The remaining targets are predominately anti-infective drug targets.

The drug target universe expands considerably if we expand our analysis to include biological targets for which medicinal chemists have developed small-molecule leads. Unlike the bioinformatics community which has developed a wealth of public databases to assemble and disseminate protein and genomic sequences, medicinal chemistry structure–activity relationship (SAR) data is not publicly available in a systematic database and is spread between com-

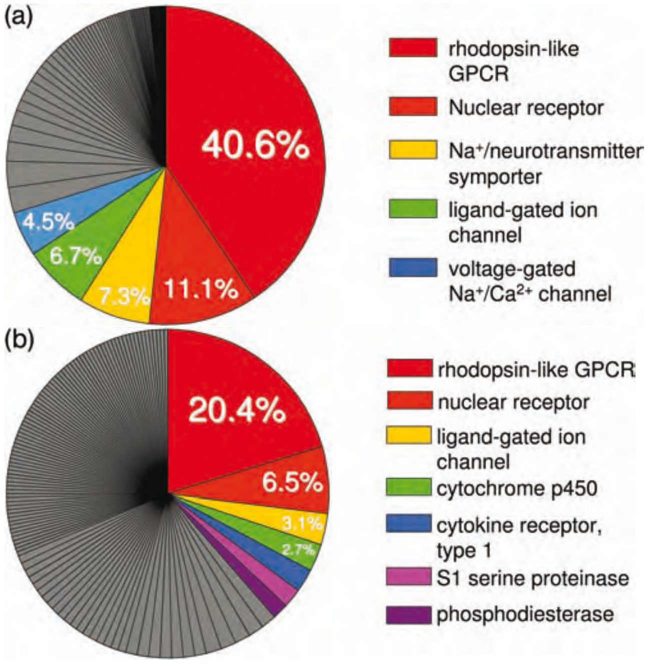


Figure 2 Molecular targets of current Food and Drug Administration approved drugs (a) by numbers of drug substances and (b) by number of drug target in gene family. Figures are derived from analysis of 1606 active ingredients (25 024 approved products). Orange Book, September 2002 (http://www.fda.gov/cder/ob).

Table 1 Molecular targets of approved drugs

Class of drug target	Species	No. molecular targets
Targets of approved NMEs	all (anti-infectives and human)	301
Targets of approved NMEs	human only	238
Targets of approved NCEs	human	**170**
Targets of approved antibodies	human	15
Targets of approved biologicals	all (anti-infectives and human)	59

pany in house data warehouse, of peer-reviewed journal articles and patents, often in formats not easily accessible to machine processing. In order to survey the universe of drug targets with known leads, Inpharmatica have created the StARLITe database of bioactive compounds by extracting structures, assays, targets and SAR from the key medicinal chemistry journals (i.e. *Journal of Medicinal Chemistry* 1980–2004, *Bioorganic and Medicinal Chemistry Letters* 1990–2004) covering 350 000 compounds and 1 275 000 assay points. The comprehensive survey of medicinal chemistry identifies 1155 targets known to have at least one drug or lead compound bind with an affinity below 10 μM, 707 of which are human molecular targets (Table 2 and Figure 3). Applying

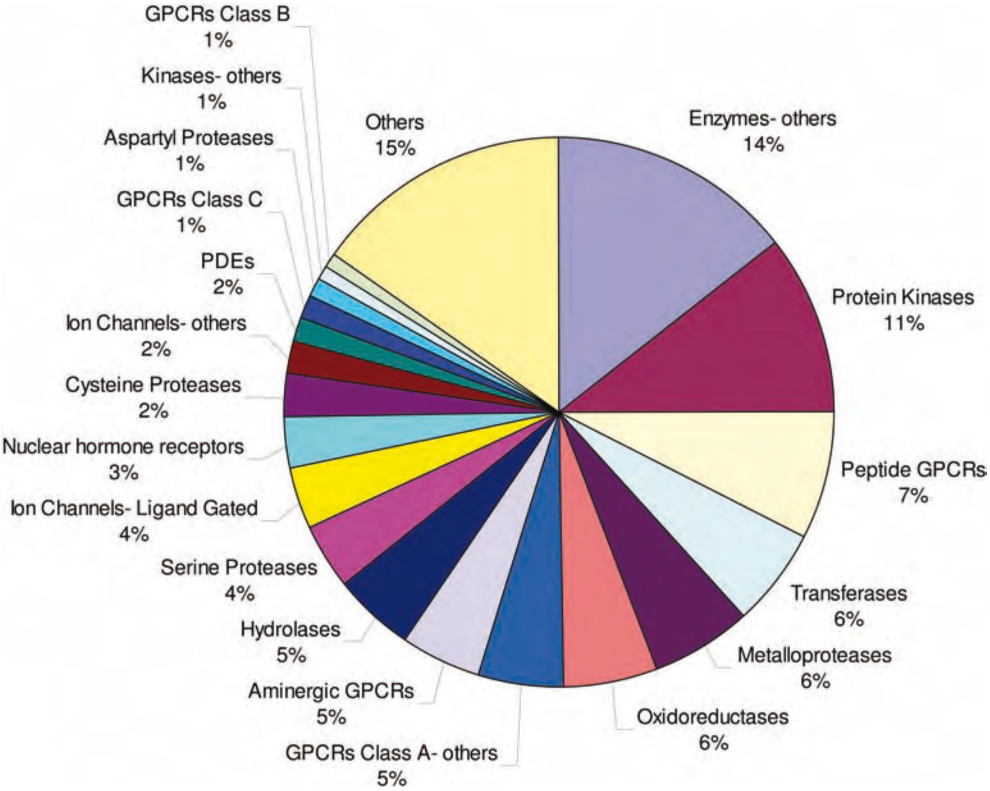


Figure 3 Gene family distribution of nonredundant human proteins with small-molecule chemical leads with binding affinities below 10 μM. Data derived from an analysis of Inpharmatica's StARLITe database.

Lipinski's criteria to the compounds in the dataset (as represented as desalted, canonical SMILES strings) reveals 587 human proteins with at least one or more compounds which complies with the "Rule of Five" with a binding affinity more potent than 10 μM, which could be unambiguously identified and assigned to a protein sequence (Figure 4). The extremely through analysis of the literature, represented in the StARLITe database, more than doubles in size the number of identified proteins with existing lead matter.

Using this larger database of drug targets which show some precedent of modulation by small-molecule leads or drugs we attempted to estimate the size of the potential druggable genome based on a homology to known drug targets. The underlying assumption in this analysis is that if one gene family member has shown the propensity to selectivity bind small-molecule modulates other members of the gene family may significant contain physicochemical and architectural properties that they are also like to bind drug-like small molecules. Proteins that have a similar sequence are generally likely to

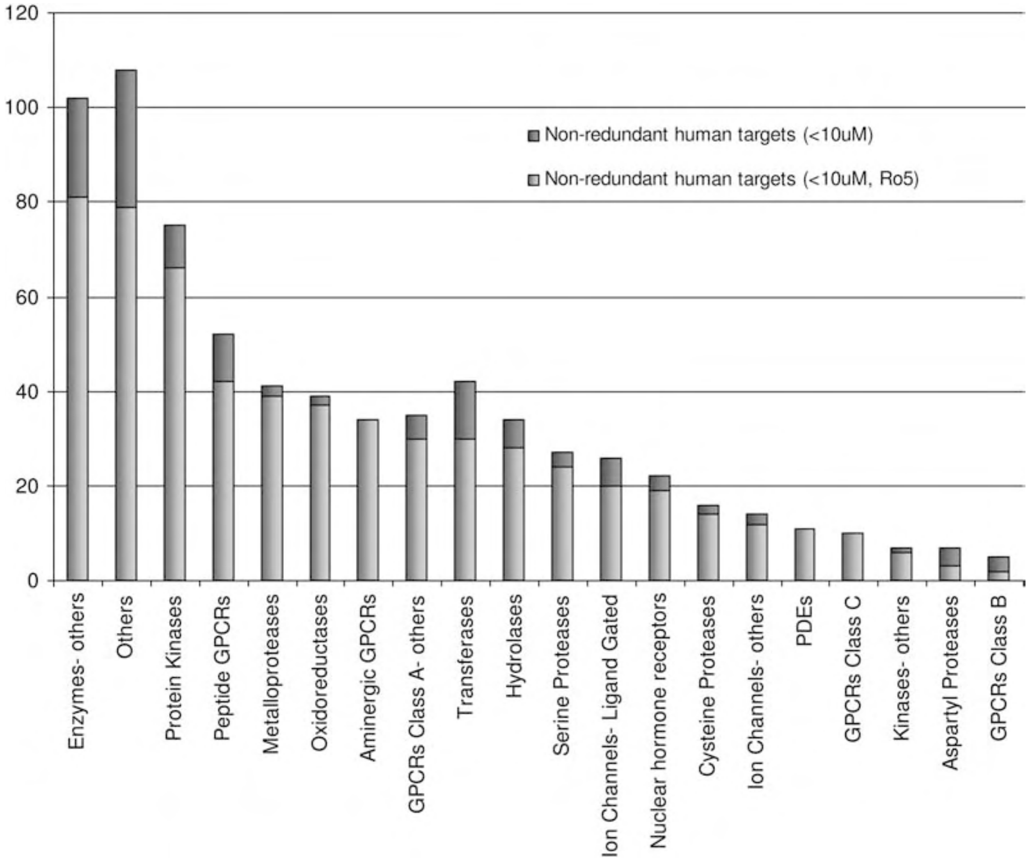


Figure 4 Proportion of targets with leads observed with at least one "Rule of Five" (Ro5) compliant compound within each gene family.

share very similar three-dimensional and perform similar or related functions. If a protein therefore has a high degree of sequence similarity to the target of a drug (or other proteins that is known to be druggable), we predict that proteins is likely to be druggable too, if we believe the binding site architecture to be conserved. Where proteins are less closely related in sequence, it is more difficult to infer druggability. Relatively small differences in the binding site of a protein could have a large impact on its ability to bind small molecules. The authors recognize that this is a simplistic assumption and is likely to overestimate the number of potential members of the predicted druggable subset of the human genome. For example, many individual members of the gene family may bind distinct ligands. The molecular recognition properties of their respective binding sites could be significantly divergent. Using the BLAST sequence alignment algorithm to search each of the sequences against the human genome, we identified 945 distinct genes that show homology to

Table 2 Molecular targets with chemical leads and tools (identified from the medicinal chemistry literature in Inpharmatica's Startlite database and unambiguously assigned to a molecular target via a protein sequence)

Gene family	Redundant ortholog targets (all species) <10 μM	"Rule of Five" redundant ortholog targets (all species) <10 μM	"Rule of Five" redundant ortholog mammalian targets <10 μM	"Rule of Five" redundant ortholog mammalian targets <10 μM	Nonredundant human targets <10 μM	"Rule of Five" nonredundant human Targets <10 μM
Aminergic GPCRs	71	71	61	61	34	34
Aspartyl proteases	10	4	9	4	7	3
Cysteine proteases	20	18	19	17	16	14
Enzymes – others	149	117	131	104	102	81
GPCRs class A – others	59	47	49	38	35	30
GPCRs class B	12	7	10	5	5	2
GPCRs class C	20	20	19	19	10	10
Hydrolases	54	44	46	37	34	28
Ion channels – ligand gated	52	42	47	37	26	20
Ion channels – others	20	18	18	16	14	12
Kinases – others	11	8	11	8	7	6
Metallo-proteases	60	56	53	50	41	39
Nuclear hormone receptors	45	33	33	26	22	19
Others	188	144	146	109	108	79
Oxido-reductases	67	63	62	58	39	37
PDEs	15	13	15	13	11	11
Peptide GPCRs	99	72	80	59	52	42
Protein kinases	101	90	87	78	75	66
Serine proteases	34	30	34	30	27	24
Transferases	68	46	57	39	42	30
Total	**1155**	**943**	**987**	**808**	**707**	**587**

the molecular targets of approved drugs at a cutoff of 30% sequence identity and E-value less than or equal to 10^{-5}. Expanding the BLAST analysis to include human proteins from the known drug-like leads from the StARLITe database identified a 2921 protein sequences within the same sequence identify cutoffs.

In addition to using a sequence homology approach, we also approached the problem of identifying the druggable subset of the human proteome using a feature-based Bayesian method.

6 Feature-based Druggability Prediction

Drug targets, be they targets of small-molecular-weight drugs or protein therapeutics, may share common sequence-based features that are not necessarily detectable by overall sequence similarity. An alternative approach to using sequence-based similarity methods is to examine the presence of sequence-based features that are enriched in drug targets compared to that of the rest of the genome. A large set of over 100 protein properties and features were calculated for each sequence in the DrugStore database such as the number of transmembrane helices, signal peptides, isoelectric point, length distribution, percentage of helical structure, antigenicity, net charge at pH 7.4, domain complexity, subcellular localization, etc. Features that were enriched in existing drug targets retained and used to construct probabilistic Bayesian models for both small-molecule druggability prediction and protein therapeutic druggability prediction. The implementation of this Bayesian probabilistic scoring allows for ranking any portfolio of targets based on their predicted druggability. The major advantage of this approach is the independence of any prior knowledge about the examined protein, or homology to precedent target families. The Bayesian models also hold the advantage of being tunable to reflect specific gene families, or drug profiles. The probabilistic models were then used to rank all sequences from human genome according to both small-molecule and protein druggability as predicted by the presence of druggable features in the protein sequence. The small-molecule model predicts 2325 gene products to be druggable with high confidence (i.e. achieving scores comparable with those of existing targets).

7 Structure-based Druggability Analysis of Protein Data Base (PDB) Structures

Following the hypothesis that druggable binding sites can be predicted *a priori* we have developed an algorithm to analyze the PDB for druggable binding sites. Actual and putative ligand-binding sites were, respectively, identified either by virtue of the presence of a ligand in the crystal structure or by analysis of the surface of the protein structure. A range of physicochemical properties of the identified binding sites and cavities were calculated from the protein structures including volume, depth, curvature, accessibility, hydrophobic surface area and polar surface area. The algorithm was trained set against a test set of 400 protein complexes binding small molecule, "Rule of Five" compliant ligands. From this analysis a decision tree was derived to predict the druggability of a binding site or cavity from calculated physicochemical properties. The decision tree predicts whether a cavity is druggable within the statistical confidence of the tree. This method has a demonstrated a 91% success rate when predicting druggability on the protein drug targets (of oral drugs as defined in Inpharmatica's Drugstore database of approved drugs). The method requires either an experimentally derived structure or a high-quality homology model. Ideally, due to the inherent flexibility of many protein-ligand binding sites a sample of multiple conformations is preferred. The method is scalable to be employed on the entire PDB (December 2004 release). By removing short peptides, 27 409 files were suitable for analysis, which was further classified into 76 322 structural domains using SCOP [23] and DISCObase of which 28% (21 522) of the structural domains were found to have at least one site predicated, to some degree, to be druggable. Due to the high redundancy in the PDB and the high number of ligand–protein complexes reduced to a nonredundant set of human targets 427 proteins were predicted to contains a druggable binding site, with 281 of these proteins having no prior known compounds or drugs developed against those targets. Structure-based druggability algorithms could be automatically applied to continuously assess the stream of novel structures determined by the structural genomics initiatives.

Combining a nonredundant set of genes from all methods outlined above:

- current targets of approved drugs
- current targets of chemical leads or chemical tools
- sequence homology to current drug targets
- sequence homology to current chemical lead targets
- feature-based sequence probability prediction
- structured-based prediction
- sequence homology to structure-based prediction

we identify a total of 3505 unique genes that are predicted with first- and second-order evidence and high confidence to encode small-molecule druggable proteins of which only 170 are the primary human targets for marketed drugs (Table 3). The results of this combined analysis concur with the previous estimated by Hopkins and Groom [14] which approximately 14% of the human genome could be inferred to be potentially druggable.

Table 3 Predictions of the size of the human druggable genome

Druggability prediction method	No. molecular targets
Targets of approved NCEs	170
Sequence homology to NCE drug targets	945
Targets of chemical leads with activities (binding affinities) below 10 μM	707
Targets of "Rule of Five" chemical leads with activities (binding affinities ≤10 μM)	587
Sequence homology to targets with chemical leads	2921
Feature-based druggability sequence probability prediction	2325
Structured-based prediction	427
Sequence homology to proteins predicted druggable by structure-based method (high confidence)	3541
Sequence homology to proteins predicted druggable by structure-based method (low confidence)	6619
Predicted druggable genome (high confidence)	**3505**

Unique druggable targets from combining drug targets, targets with leads, homology to drug/lead targets and structure-based prediction.

8 How Many Drug Targets are Accessible to Protein Therapeutics?

If our explorations of the proportion of the protein targets expressed by the human genome accessible to modulation by high-affinity, drug-like small molecules is limited, how much larger is the universe for drug targets if we expand our investigations to include targets of protein therapeutic such as antibodies and recombinant biologicals? At the time of writing, approved antibody therapeutics were known to act on 15 human targets, whilst in total all biological drugs in the pharmacopoeia currently work via 59 modes of action. Due to the inherently lower toxicity observed for fully humanized antibodies and the rising rate of biological approvals, it has been argued that antibodies may soon overtake NCE approvals [2]. Interestingly, it has also been observed by studying rates of attrition that antibodies acting against novel modes of action often shown a higher chance of success in phase II clinical studies than small-molecule drugs acting on precedent mechanisms [26, 29, 37]. Thus, we attempted to estimate how many targets are accessible to biological drugs as the targets of antibody therapies. Other criteria, such as antigenicity, are also important in developing inhibitory antibodies. However, these have not been considered in this analysis, as they are not common to both antibody and other protein drugs.

In order to estimate the number of genes expressing products that could be accessible to antibody therapeutics we assume that proteins are required to be located in the extracellular matrix. We assume that extracellular location is the common characteristic of secreted and transmembrane of proteins. Where the extracellular location is known, this is often included in Swiss-Prot and Gene Ontology (GO) [4] database annotation for the protein. Secreted proteins can be predicted by the presence of a signal peptide, whilst transmembrane domains can be identified by sequence property prediction. Analysis reveals 1384 genes predicted to encode secreted proteins with high confidence (i.e. predicted by multiple different methods). If the confidence level is lowered (i.e. signal peptide predicted by single method), 6560 genes are predicted to potentially be secreted. Our transmembrane analysis reveals that 973 genes are predicted, by multiple methods, to have transmembrane domains and be located inside the plasma membrane. This number increases by 1407 genes which are predicted to be plasma membrane proteins only by a single method. Combining these results together we identify the total number of extracellular proteins, with high confidence, to be expressed by 2287 genes. The study was extended to identify proteins that contain feature similar to the current set of biological drug targets using the Bayesian probabilistic featured-based algorithm discussed above. Trained on the existing set of biological drug targets 1637 gene products were predicted to be druggable *via* biological therapeutics with high confidence (i.e. achieving scores comparable with those of existing protein targets). Therefore, the total number of genes predicted to encode protein therapeutic druggable proteins is 3258 genes equivalent to 13% of the gene in the human genome (Table 4).

Table 4 Predictions of the number of genes in the human genome accessible to protein therapeutics (recombinant soluble proteins and antibodies)

Druggability prediction method	No. of molecular targets
Targets of approved antibodies	15
Targets of approved biologicals	**59**
Secreted protein (high confidence)	1384
Secreted proteins (low confidence)	6560
Transmembrane predictions (high confidence)	973
Transmembrane predictions (low confidence)	1407
Unique, combined transmembrane and secreted predictions (high confidence)	2287
Feature-based biological target sequence probability prediction	1637
Total unique genes predicted to be accessible via biological therapeutics	3258

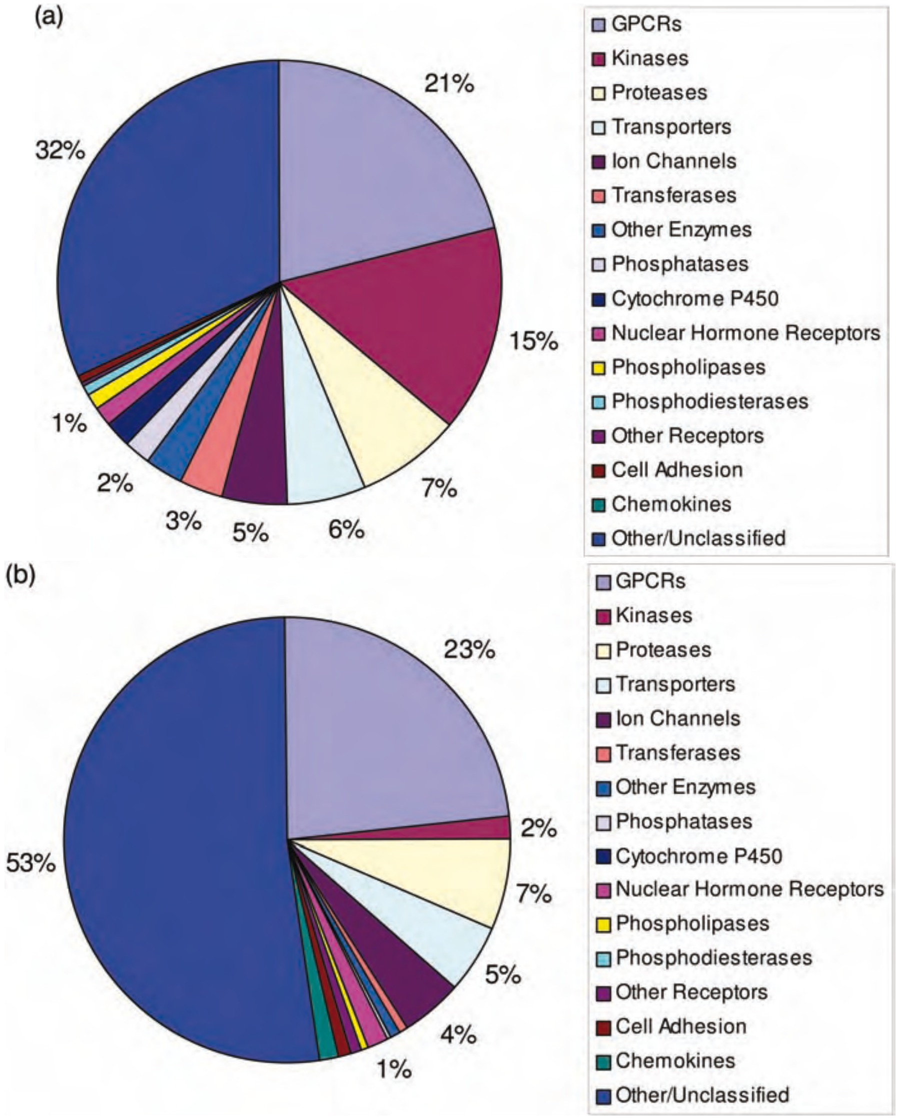


Figure 5 Gene family distributions: (a) small-molecule druggable genome and (b) protein therapeutics.

9 Conclusions

From a comprehensive survey of the medicinal chemistry literature, and by combining a variety of methodologies, sequence homology, structure- and feature-based, we have identified approximately 3500 genes in the human genome that are predicted to be accessible to modulation by high affinity

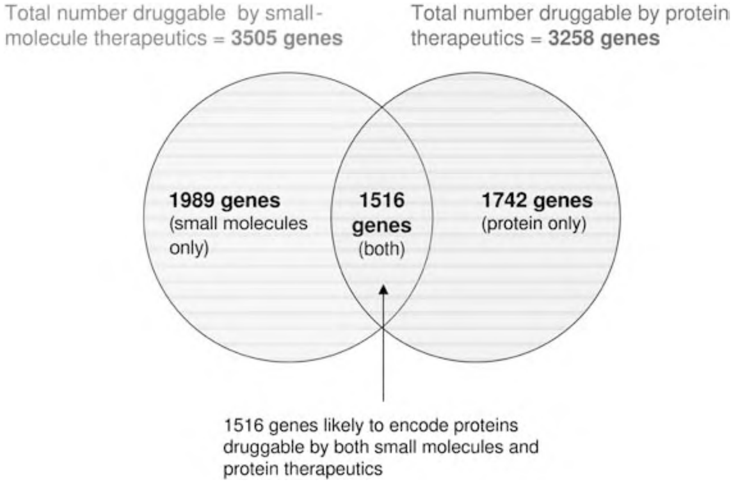

Total number druggable by small-molecule therapeutics = 3505 genes
Total number druggable by protein therapeutics = 3258 genes
1989 genes (small molecules only)
1516 genes (both)
1742 genes (protein only)
1516 genes likely to encode proteins druggable by both small molecules and protein therapeutics

Figure 6 Overlap of antibody and small-molecule druggable universes.

drug-like small molecules: approximately 14% of the human genome. Of the approximate 3500 human druggable genes, small-molecule chemical tools or leads (with binding affinities equal or more potent that 10 μM) have already been identified that act on 707 of these and 170 are the primary targets for approved, small-molecule drugs. Whilst there may be many more proteins expressed by the human genome that may be discovered to be modulated by small-molecule tools or drugs, the proteins identified as belong to the subset known as the "druggable genome" represent those targets we can readily predict have a higher confidence of discovering a small-molecule chemical tool than the remaining genes in the genome. Since it was first proposed that the various physicochemical constraints on drug-like chemicals would reduce the available targets space it has been suggested that the accessible drug target space may expand considerable with the application of biological drugs such as fully humanized antibodies. To date, approved protein therapies act via about 59 human targets, 18 of these are targeted by marketed antibodies. With the commercialization of recombinant protein production the number of biological drugs receiving approval and being studied in the clinic is steadily rising. Several commentators predict that the rise of antibody therapies may challenge the premier position of small-molecule chemical entities as the dominant technology of medicinal chemistry [2]. Our analysis of the proposition of the genome potentially accessible to modulation by protein therapeutics, such as antibodies is around 13%, with 3258 genes predicted to encode proteins potential druggable via protein therapeutic proteins. Interestingly, 70% of all the drug targets are also predicted to be accessible to modulation by antibody therapy. Indeed, if we expand the analysis to compare the overlap between the antibody-accessible druggable genome and the “small-molecule druggable” genome, 1516 genes are predicted to encode proteins druggable by both small molecules and protein therapeutics – approximately 45% of our current estimate of the small-molecule druggable genome (Figures 5 and 6).

Acknowledgments

We would like Colin Groom (UCB Celltech, Cambridge, UK) for his long-standing contribution to this work. We also sincerely thank Edith Chan (Inpharmatica, London), Robin Spencer (Pfizer, Groton), Lee Beeley (Pharmamatters, Ramsgate) and Jonathan Mason (Pfizer, Sandwich) for their helpful discussions in this development of this work.

References

36 The Molecular Basis of Predicting DruggabilityReferences

1 Ajay, A., W. P. Walters and M. A. Murcko. 1998. Can we learn to distinguish between “drug-like” and “nondrug-like” molecules? J. Med. Chem. **41**: 3314–24.

2 Arlington, S., S. Barnett, S. Hughes and J. Palo. 2002. *Pharma 2010: The Threshold of Innovation.* IBM Business Consulting Services, London.

3 Consortium, G. O. 2001. Creating the gene ontology resource: design and implementation. Genome Res. **11**: 1425–33.

4 Davies, K. 2002. Cracking the “druggable genome”. Bio-IT World: http://www.bio-itworld.com/archive/100902/firstbase.html.

5 Drews, J. 2000. Drug discovery: a historical perspective. Science **287**: 1960–4.

6 Drews, J. 1996. Genomic sciences and the medicine of tomorrow. Nat. Biotechnol. **14**: 1516–8.

7 Drews, J. and S. Ryser. 1997. Classic drug targets. Nat. Biotechnol. **15**: 1318–9.

8 Egan, W. J., W. P. Walters and M. A. Murcko. 2002. Guiding molecules towards drug-likeness. Curr. Opin. Drug Discov. Dev. **5**: 540–9.

9 Ertl, P., B. Rohde and P. Selzer. 2000. Fast calculation of molecular polar surface area as a sum of fragment based contributions and its application to the prediction of drug transport properties. J. Med. Chem. **43**: 3714–7.

10 Golden, J. 2003. Towards a tractable genome: knowledge management in drug discovery. Curr. Drug Discov. **3(2)**: 17–20.

11 Golden, J. B. 2003. Prioritizing the human genome: knowledge management for drug discovery. Curr. Opin. Drug Discov. Dev. **6**: 310–6.

12 Hajduk, P. J., J. R. Huth and S. W. Fesik. 2005. Druggability indices for protein targets derived from NMR-based screening data. J. Med. Chem. **48**: 2518–25.

13 Hopkins, A. L. and C. R. Groom. 2002. The druggable genome. Nat. Rev. Drug Discov. **1**: 727–730.

14 Hopkins, A. L. and C. R. Groom. 2003. Target analysis: *a priori* assessment of druggability. Ernst Schering Res. Found. Workshop **42**: 11–7.

15 Kuntz, I. D., K. Chen, K. A. Sharp and P. A. Kollman. 1999. The maximal affinity of ligands. Proc. Natl Acad. Sci. USA. **96**: 9997–10002.

16 Lajiness, M. S., M. Vieth and J. Erickson. 2004. Molecular properties that influence oral drug-like behavior. Curr. Opin. Drug Discov. Dev. **7**: 470–7.

17 Lander, E., L. M. Linton, B. Birren, et al. 2001. Initial sequencing and analysis of the human genome. Nature **409**: 860–921.

18 Lipinski, C. A. 2000. Drug-like properties and the causes of poor solubility and poor permeability. J. Pharmacol. Toxicol. Methods **44**: 3–25.

19 Lipinski, C. A., F. Lombardo, B. W. Dominy and P. J. Feeney. 1997. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. Adv. Drug Del. Rev. **23**: 3–25.

20 Muegge, I. 2003. Selection criteria for drug-like compounds. Med. Res. Rev. **23**: 302–21.

21 Muegge, I., S. L. Heald and D. Brittelli. 2001. Simple selection criteria for drug-like chemical matter. J. Med. Chem. **44**: 1841–6.

22 Murzin, A. G., S. E. Brenner, T. Hubbard and C. Chothia. 1995. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J. Mol. Biol. **274**: 536–40.

23 Orth, A. P., S. Batalov, M. Perrone and S. K. Chanda. 2004. The promise of genomics to identify novel therapeutic targets. Expert Opin. Ther. Targets **8**: 587–96.

24 Overington, J. 2002. Prioritizing the proteome: identifying pharmaceutically relevant targets. Drug Discov. Today **7**: 516–21.

25 Pavlou, A. K. and J. M. Reichert. 2004. Recombinant protein therapeutics – success rates, market trends and values to 2010. Nat. Biotechnol. **22**: 1513–9.

26 Podlogar, B. L., I. Muegge and L. J. Brice. 2001. Computational methods to estimate drug development parameters. Curr. Opin. Drug Discov. Dev. **4**: 102–9.

27 Proudfoot, J. R. 2002. Drugs, leads, and drug-likeness: an analysis of some recently launched drugs. Bioorg. Med. Chem. Lett. **12**: 1647–50.

28 Reichert, J. M. 2005. Protein therapeutic success rates increase with biotech advances. Tufts Center for the Study of Drug Development Impact Report **7**.

29 Russ, A. P. and S. Lampel. 2005. The Druggable Genome: an update. Drug Discov. Today **10(23–24)**: 1607–10.

30 Veber, D. F., S. R. Johnson, H. Y. Cheng, B. R. Smith, K. W. Ward and K. Kopple. 2002. Molecular properties that influence the oral bioavailability of drug candidates. J. Med. Chem. **45**: 2615–23.

31 Venter, J. C., M. D. Adams, E. W. Myers, et al. 2001. The sequence of the human genome. Science **291(5507)**: 1304–51.

32 Vieth, M., M. G. Siegel, R. E. Higgs, I. A. Watson, D. H. Robertson, K. A. Savin, G. L. Durst and P. A. Hipskind. 2004. Characteristic physical properties and structural fragments of marketed oral drugs. J. Med. Chem. **47**: 224–32.

33 Walters, W. P., Ajay and M. A. Murcko. 1999. Recognizing molecules with drug-like properties. Curr. Opin. Chem. Biol. **3**: 384–7.

34 Walters, W. P. and M. A. Murcko. 2002. Prediction of "drug-likeness". Adv. Drug Deliv. Rev. **54**: 255–71.

35 Wang, J. and K. Ramnarayan. 1999. Towards designing drug-like libraries: a novel computational approach for prediction of drug feasibility of compounds. J. Comb. Chem. **1**: 524–33.

36 Windhoven. 2005. Know they R&D enemy: the key to fighting attrition. In Vivo **19(1)**: 59.

Note: This chapter is based on an article submitted for publication in "Chemical Biology", edited by Stuart L: Schreiber, Tarun M. Kapoor and Günter Wess (ISBN 978-3-527-31150-7).

Part 9
Comparative Genomics and Evolution of Genomes

37
Comparative Genomics

Martin S. Taylor and Richard R. Copley

1 Introduction

Comparative genomics is an approach that uses the signal of past selection as a highly sensitive assay for function in genome sequences. Unlike experimental approaches, it does not require a prior hypothesis of that function. The realization that comparative sequence analysis is crucial to understanding the functions encoded in the human and other genomes is driving a major comparative sequencing effort. The fruits of this labor are a rapidly expanding number of whole-genome sequences and new computational methods to analyze these data in efficient and meaningful ways.

In this chapter, we introduce the concepts and techniques of comparative genomics; in doing so we build on the foundations laid in earlier chapters, particularly Chapter 3 on sequence alignment and Chapter 4 on phylogenetics. In general terms, the approaches we describe can be applied to any collection of organisms, but our emphasis here is primarily on questions of relevance to human genomes. We begin, in Section 2, by presenting an overview of genome structure and content, providing a context for the subsequent discussions. We then introduce the concepts of natural selection, homology and phylogenetic distance that underlie comparative genome analyses in Section 3. In Section 4, we consider the types of questions that can be addressed and the strategies that can be employed to address them. We also consider the availability and accuracy of genomic sequence data. In Section 5, we introduce the three main technical challenges of comparative genomic sequence analysis – genomic sequence alignment, the visualization of sequence relationships and detecting the signal of selection. We review the methods employed to meet

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

these challenges and discuss the most popular, and the most promising new tools. In Section 6, we illustrate the utility of comparative genomic studies with recent applications that have given new insights into human biology. Finally, in Section 7, we highlight some resources that are likely to have a profound impact on future comparative genomic studies and identify future research challenges.

2 The Genomic Landscape

The human genome is approximately 3 200 000 000 (3.2 Gb) nucleotides long [50]. At first sight, a monotonous repetition of A, T, C and G representing the four nucleotides of DNA, it is in fact a diverse and still in many ways mysterious landscape. Of the total 3.2 Gb, 2.85 Gb have been sequenced to high accuracy [50]; the remainder are largely attributable to heterochromatic regions (centromeres and telomeres) that are highly repetitive and refractory to current sequencing technology.

Proteins are often thought of as the principal functional product of a genome. Consequently, protein-coding sequences are the first place screened for disease-associated mutations and functionally significant polymorphisms. The human genome encodes approximately 22 000 protein-coding genes (http://www.ensembl.org), although the total diversity of proteins produced is likely to be several times this thanks to alternate transcription initiation and processing [19, 67]. However, it appears that this protein-coding sequence accounts for less than 1.5% of the human genome sequence [59]. The situation is similar in rodents [40, 114], other mammals [63] and, to varying degrees, other vertebrates [4, 46]. These estimates of coding sequence content appear to be robust, they are based on the integration of comparative data [90] and transcript evidence [89], and they are also consistent with extrapolation from the detailed investigation of targeted regions [75]. This finding does of course raise the question "What is the function of the remainder of the genome?" and even questions whether the majority of a vertebrate genome is subject to any selective constraint.

A protein is translated from an mRNA which contains a contiguous protein-coding "open reading frame", flanked by stretches of noncoding sequence. In eukaryotes, the genomic DNA that provides the template for mRNA synthesis is often found as a series of collinear but discrete segments (*exons*), interrupted by *introns*. Although transcribed, the introns are spliced from the mRNA prior to protein synthesis. This is a marked difference from the situation in prokaryotic organisms (Chapter 41), where the mRNA template is typically uninterrupted. Particularly in higher eukaryotes such as mammals, the length of introns often far exceeds that of exons. In addition to the sequences that are transcribed into RNA, a gene will have an associated core-promoter region immediately upstream of the transcribed region. The core-promoter region acts to drive transcription and consequently expression of the gene. However, it is common, especially in higher eukaryotes that there are other sequences, potentially distantly located [61], that are also crucial to the appropriate regulation of gene expression.

Pseudogenes are the remnants of functional genes that have apparently lost the ability to perform any useful function. Often, this means that they look obviously like protein-coding genes, but that they have accumulated changes such as in-frame stop codons that would prevent them producing a functional protein. Pseudogenes can arise through duplication where one copy of a gene retains and continues to perform the original function, whereas the other is redundant so free to accumulate changes that disrupt its function without detriment to the fitness of the organism. Alternately, the function of the gene may no longer be required – the human *GLY1* pseudogene possibly provides such an example, the ancestral gene acting in threonine catabolism, but for which the only human copy contains multiple in-frame stop codons [31], again the selective constraint is removed. In both of these cases the gene becomes a nonprocessed pseudogene.

A major component of the human and many other higher eukaryotic genomes is sequence derived from *interspersed repetitive elements (IRE)* such as endogenous retroviruses, retrotransposons and DNA transposons. At least 45% of the human genome is identifiably derived from IREs [59], although this almost certainly underestimates their true contribution as older, more divergent repeat-derived sequences are unlikely to be identified. These elements are often considered "junk" DNA and rarely have organism-level biological functions been attributed to them, although a small number of exceptions are known [52, 58]. It is interesting to note that some vertebrate lineages, most notably that of the pufferfish, are almost devoid of such IRE-derived sequence and have a genome approximately 8-fold smaller than the human despite encoding a similar number, or possibly slightly more, protein-coding genes [4].

Processed pseudogenes are a consequence of a genome rich in repetitive elements, specifically those that replicate and transpose through the reverse transcription of RNA into DNA. Occasionally, rather than reverse transcriptase, the enzyme responsible for reverse transcription, driving the replication of an IRE, it will reverse transcribe the mRNA of a gene. The result is a processed pseudogene – the copy of an mRNA integrated into the genome. These are distinct from the pseudogenes discussed above as they bare the hallmarks of transcript processing, such as the removal of introns and $3'$ polyadenylation [119]. Processed pseudogenes are often incomplete at the $5'$ end – a consequence of reverse transcriptase reading the $3'$ end of the mRNA first. Some genes such as those encoding the ribosomal proteins are particularly susceptible to generating new processed pseudogenes [119], probably reflecting in part the level germline transcript of the gene.

In probably every measure that has been made of the human genome sequence, it has been found to be far from homogeneous. We have already seen the distinction between heterochromatic regions that appear to perform roles in the packaging and segregation of chromosomes. Throughout the rest of the genome (the euchromatic regions), there is considerable variation in gene density (the number of genes per unit sequence), IRE content, nucleotide and dinucleotide frequency, and the observed rates of genetic recombination, nucleotide substitution, insertions and deletions. Many of these attributes have been found to covary across the genome [40, 45, 101], but currently the bases of their interrelationships are not well understood. Of particular relevance to comparative genomic studies is the fluctuation of substitution, insertion and deletion rates across the genome [117], which suggest there may be regional variation in the rate at which mutations occur. At least in rodents, the scale of this variation is of the order of 1 Mb, so that the substitution rates for two neutrally evolving regions of sequence are highly correlated if they lie within this distance of each other, but the correlation decreases rapidly with increasing genomic distance [38].

The rate of sequence mutation is not only dependent on the large-scale region of a genome, but also on the sequence and composition of neighboring nucleotides [45, 108]. For example, tandemly repeated sequences and mononucleotide tracts are prone to insertion and deletion mutation [108]. The epigenetic methylation of cytosine nucleotides when they are located directly upstream of a guanine (CpG) is a common occurrence in mammalian genomes and, to a lesser extent, in other metazoa [8]. This nucleotide modification has had a major influence in shaping mammalian genomes. Thanks to a quirk of biochemistry, a methylated C can mutate to T at a much higher frequency than all other nucleotide substitutions can occur. As a result, CpG dinucleotides are grossly under-represented across the majority of the human genome, relative to chance expectation given the frequency of C and G nucleotides (approximately 20% of the expected frequency [59, 104]) and CpG mutation rates tend to be substantially higher than those of other dinucleotides. However, within specific islands of sequence (commonly known as CpG islands), CpGs are not methylated, at least in the germline [9], so are not under-represented. CpG islands are often associated with the $5'$ end and promoters of some genes [9], so represent sequences that are often of particular interest in comparative genomic studies.

Segmental duplications are a genomic feature that can often cause problems for sequence assembly such that they are frequently overlooked. These are large (typically 5 kb is taken as a minimum threshold in their definition) tracts of sequence that occur multiple times in a genome, often as tandem repeats. These duplicated regions can contain whole genes or even multiple genes. Recent segmental duplications will share a high degree of nucleotide identity and are likely to be polymorphic in the population, and the mechanism of segmental duplication provides a rapid means of divergence between species [60, 80].

3 Concepts

The replication of DNA is imperfect – new mutations are continually arising with each generation. In the absence of natural selection, the eventual fate of a new mutation will be determined by genetic drift, i.e. chance fluctuations in frequency that result from sampling a finite population. For most mutations this will result in their loss from the population, but some will drift to fixation. As this is a random process, any observed sequence changes can be considered an unbiased sample of all mutations that occurred. However, natural selection disrupts this unbiased sampling of mutations.

Through the cumulative action of past evolution, most functional DNA is expected to have attained a sequence that is near-optimal for its environment. Consequently, mutational changes are likely be detrimental, i.e. to result in a departure from the optima and removed by purifying selection. As a result, functionally important sequences are expected to accumulate fewer mutational changes than neutrally evolving DNA, so functional regions of two sequences diverged from a common ancestor are expected to be more similar than nonfunctional regions. Local regions of sequence similarity resulting from selective constraint are often referred to as a phylogenetic footprint [105].

At the opposite end of the selective spectrum is diversifying (positive) selection. Following environmental change, an existing sequence may no longer be optimal and new mutations could provide a selective advantage, in which case they are likely to increase in frequency in the population, as it responds to the change in selective pressure. Diversifying selection can lead to changes accumulating at a faster rate than in neutrally evolving sequence. There are instances such as sexual selection and host– pathogen arms races where there is sustained selective pressure for diversification [81], but in the majority of cases a period of diversification will be both preceded and succeeded by longer periods of purifying selection. As such, diversifying selection can be difficult to unambiguously identify and the majority of comparative studies outside protein-coding sequences currently focus on the identification of purifying selection. For an in-depth discussion of genetic drift, selection and the influence of population size, see Ref. [65].

Both purifying and diversifying selection result in a departure from the neutral rate of sequence evolution; this departure is diagnostic and can be considered the signature of selection. Natural selection can only act on genetic variation that manifests as phenotypic differences between individual organisms of a population. It is a stringent filter – even a 0.001% reduction in fitness will result in a polymorphism being efficiently removed from most mammalian populations [83, 86]. Therefore, the signature of selection defines a sequence as significantly contributing to the biology of the organism. As we have discussed (Section 2), vertebrate and many other higher eukaryotic genomes are dominated by sequences that appear to have no biological function. This means that although the human genome is approximately 3.2 Gb in size [59, 112], most of the biological functions, and, consequently, disease-associated polymorphisms and biological insight are concentrated into as little as 0.16 Gb of sequence [40, 64] (Section 6.1). Comparative genomics provides a means of identifying that rich vein of functional sequence and, unlike laboratory-based approaches, it does so without requiring prior assumptions of what that function may be.

The rate of sequence evolution is measured from an alignment (Chapter 3) between sequences that have diverged from a common ancestor, i.e. they are homologous. If the point of divergence for two homologous sequences was a speciation event, then they are referred to as orthologs. Otherwise, they are paralogs of one another. The distinction between *orthology* and *paralogy* is important for two reasons. (i) Orthologs are more likely than paralogs to have conserved the same function since divergence because the processes giving rise to paralogs such as intra-genome duplication and horizontal gene transfer provide an opportunity for functional diversification through the relaxation of selective constraint [43]. (ii) When comparing multiple orthologs loci between the same range of organisms, a common phylogenetic relationship and divergence times can be assumed for all of the loci, enabling direct comparison between loci. No such assumptions can be made for comparisons involving paralogous loci. For these reasons the majority of studies are based on alignments of orthologous sequences.

Phylogenetic scope, a term introduced by Cooper and coworkers [23], defines the range of organisms being considered in an analysis, denoted by their most recent common ancestor. For example, a study involving sequences from zebrafish, chicken, frog, mouse and human would be vertebrate in scope, whereas one looking at human, chimp and macaque is primate in scope. The phylogenetic scope of a study must be matched to the biological questions being asked. In general, more closely related species are more likely to have similar biology than distantly related species. The *Sonic hedgehog* gene discussed later (Section 6.3) provides a good example of the potential pitfalls of an inappropriate phylogenetic scope.

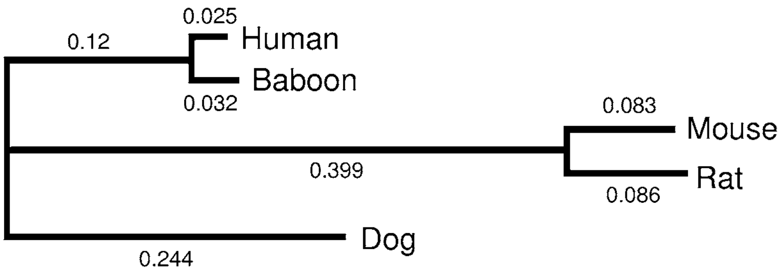


Figure 1 Phylogenetic tree showing branch lengths. An unrooted tree with branch lengths derived from nucleotide substitution rates of anonymous aligned sequence in the greater cystic fibrosis transmembrane conductance regulator gene (**CFTR**) region. Individual branch lengths are shown on each branch segment.

The number of expected differences between sequences has important implications for the utility of a particular sequence in a comparative analysis and how the analysis should be performed. It is useful then to have some standard measure of the expected degree of sequence divergence. For orthologous sequences, a widely used measure has been divergence time in millions of years, estimated through the integration of fossil records and molecular data. The greater the divergence time, the greater the number of changes that are likely to have accumulated. However, these date estimates vary wildly with the methods used and assumptions made, e.g. the divergence between rodent and primate lineages has been estimated as occurring between 75 and 121 million years ago [40,41,114].

A more useful measure for comparative genomics analysis is that of branch length, sometimes simply referred to as distance. The concept of branch length is introduced in the chapter on phylogenetic reconstruction (Chapter 4) where it denotes the number of mutational changes per unit of sequence, e.g. substitutions per nucleotide, deletions per amino acid or inversions per kilobase. The most useful and widely used measure when considering comparative genomics is that of substitutions per nucleotide, as it is readily calculated (Chapter 4) and is reasonably robust to alignment methodology. As a measure it also relates directly to the amount of information present in aligned sequences and also how accurate an alignment between those sequences is likely to be (see below). For the phylogenetic tree shown in Figure 1, the total branch length between human and mouse is $D = 0.63$ substitutions per site in neutrally evolving sequence, calculated by summing branch lengths between the human and mouse terminal nodes (0.025 + 0.12 + 0.399 + 0.083). It should be noted that branch length is often not the same as the sum of sequence differences, as the methods used to calculate substitution rates typically take into account the likelihood of multiple changes at the same site.

In theory, the power of a study to discriminate non-neutral from neutral evolution is proportional to the total divergence (branch length) of the analysis, in the case of Figure 1, this would be the sum of each value shown on the tree (total 0.989). Under the simplest scenario of identifying selective constraint, one is evaluating the likelihood that a segment of nucleotides has remained unchanged by chance, given an expected neutral rate of evolution D. For small values of D, we can use the Poisson distribution (e^{-D}) to approximate the probability that a neutrally evolving site will be unchanged [23,30]. For a human:mouse alignment with $D = 0.63$, there is a 53% likelihood that a neutral site will be unchanged by chance.

In practice, a pairwise alignment between orthologous sequences cannot discriminate selective constraint from neutral evolution for a single nucleotide position. Rather, a region of consecutive nucleotides is evaluated collectively. The size of region necessary to identify selective constraint scales inversely with the value of D for the analysis [30]. A simple way to increase the sensitivity of an analysis, to detect shorter or less conserved sequences, is to compare more distantly related sequences. Unfortunately, there are two important caveats to this strategy. (i) The more diverged sequences are, the less accurate the alignments are between them [88], so constrained sequences may be missed at the alignment stage rather than in the analysis of the alignment. (ii) The issue of phylogenetic scope – diverged species are less likely to share biological functions or be subject to similar constraints.

An alternative approach for increasing total D of an analysis is to include more sequences through multiple alignment. Based on the branch-length values in Figure 1, a comparison of human and mouse has $D = 0.63$, but adding rat as a third species increases total D to 0.72. When calculating total D for an analysis, each unique section of branch is counted only once, so rat only adds $D = 0.086$ to the total analysis, considerably more power could be added by using dog instead of, or in addition to, rat as it would contribute $D = 0.244$ of unique branch length. A further advantage to increasing comparisons from pairwise to multiple sequences is that it allows the direction of mutational changes to be resolved, such as the discrimination of insertion from deletion and the ability to assign changes to a specific lineage [40].

Alignment of closely related pairs of sequences such as human–chimp or human–macaque orthologous regions ($D = 0.009$ and 0.052, respectively [70]) are of little use for phylogenetic footprinting studies (Figure 2). However, extending the approach described above to the alignment of many such similar sequences can in theory provide sufficient total D to usefully detect selective constraint [23,30]. As the sequences are closely related, their alignment should be highly accurate, covering most nucleotides [88], and the phylogenetic scope is narrow, so little functional divergence is expected. This paradigm, know as *phylogenetic shadowing* [11], represents an ideal combination of attributes for comparative genomic studies. Using phylogenetic shadowing, Boffelli and coworkers [11] were able to demonstrate the identification of constrained sequences specific to primates and showed that as few as four to eight well-chosen genomes could capture much of the information present in deeper alignments of up to 17 primate sequences. The principal limitation is the need for multiple closely related, orthologous sequences (Section 3).

4 Practicalities

4.1 Available Genomic Sequences

At the turn of the millennium, comparative genomic projects in vertebrates involved the laboratory-based identification of homologous regions and their sequencing [26], prior to any comparative analysis. This situation has changed markedly, with an extremely high-quality reference human genome sequence in hand [50], and high-quality draft sequences from mouse and rat [40, 114]. The target for all three of these genomes is "finished" sequence, highly accurate and completely contiguous. Finished sequence is the "gold standard" and the ideal for comparative analysis. Unfortunately, the production of finished vertebrate sequence currently demands considerable human time and skill, and is correspondingly expensive. In contrast, a well-designed whole-genome shotgun sequencing and assembly project [115] can be largely automated at every stage. As a result of these economics, the majority of eukaryotic whole-genome sequencing projects now being undertaken have adopted a purely whole-genome shotgun strategy (Chapter 2), producing "draft" assemblies with no finishing step planned for the foreseeable future.

Draft assemblies have been produced from multiple other vertebrates including chicken (*Gallus gallus* [46]), dog (*Canis familiaris* [63]), zebrafish (*Danio rerio*), frog (*Xenopus tropicalis*), macaque (*Macaca mulatta*), chimpanzee (*Pan troglodytes* [22]) tiger-pufferfish (*Takifugu rubripes* [4]), domestic cattle (*Bos taurus*), rabbit (*Oryctolagus cuniculus*), armadillo (*Dasypus novemcinctus*), African elephant (*Loxodonta africana*), opossum (*Monodelphis domestica*), medaka (*Oryzias latipes*) and freshwater pufferfish (*Tetraodon nigroviridis* [51]). This list is expanding at an accelerating rate, driven largely by the realization that sequence comparisons between multiple vertebrate genomes is crucial to understanding the structural and functional components encoded in the human genome [21].

Whole-genome assemblies, rather than individual clone sequences, now provide the primary resource of genomic sequence for most comparative analyses in the vertebrate scope, and there is a similar situation for biologists focusing on prokaryotes, viruses, fungi, plants, nematodes and insects, with at least draft status sequence available for over 1000 genomes. However, the quality and completeness of sequences should be considered when undertaking an analysis. For finished sequence, the accuracy is expected to be high with less than one nucleotide error per 100 000 nucleotides and fewer than one insertion/deletion error per 200 000 nucleotides, the vast majority of which are located in tandemly repetitive sequence [50], and there should be no gaps in sequence coverage. The quality of draft sequences depends to a large degree on the depth of coverage. With 8-fold coverage (every base sequenced on average 8 times), a whole-genome shotgun sequencing project can produce a high-quality sequence with good long-range ordering of sequences [77]. As coverage is reduced, the rate of all types of error increase; in particular, there is a rapid reduction in sequence contiguity [116].

Even in high-quality and "finished" genomic sequences, there is still a chance of mis-assembly, especially in regions rich in repetitive elements. However, a more common issue is that of segmental duplication (Section 2) where very recently duplicated regions, that may encompass several genes, cannot be reliably discriminated during normal assembly procedures resulting in the collapse of multiple duplications into a single sequence [98]. Efforts are currently being made to identify and resolve these problematic regions [96]; however, it has become apparent that the copy number of high-identity (above 97%) segmental duplications is often polymorphic in the human population, diverges rapidly between species [20] and may be associated with disease susceptibility [32]. A further consideration is that the small number of differences between segmental duplicates will appear as polymorphisms in almost all assays, having potentially disruptive effects on genetic studies. For these reasons it is often prudent to check for indications of segmental duplication such as the "WSSD" and "Segmental Dups" tracks from the University of California at Santa Cruz (UCSC) Genome Browser (http://genome.ucsc.edu) prior to investigating a new region.

Considerations of sequence quality and coverage are set to become more important as the emphasis of genome sequencing continues its shift from high-accuracy sequencing to sampling more genomes, but with lower individual coverage. As discussed above (Section 3), an optimal strategy for the identification of constrained sites is to analyze sequence from many closely related genomes to achieve a large total branch length. The cost of sequencing one genome to 8 times is almost the same as eight genomes once, and there is then a trade-off between high quality sequence and maximizing the number and diversity of sequenced genomes. Margulies and coworkers [70] have explored this trade-off with both real and simulated data, demonstrating that as little as 2 times shotgun, although insufficient to produce a good-quality assembly, can be useful in the identification of constrained sequences by directly aligning reads to more completely sequenced genomes.

The National Human Genome Research Institute (NHGRI; http://www.genome.gov) has adopted this strategy of many genomes at low coverage and is currently coordinating the low-coverage sequencing of 16 additional mammalian genomes, selected to maximize total branch length for comparative analysis. The full list of organisms, target sequence coverage and progress in sequencing can be monitored online (http://www.genome.gov/10002154). Based on the equations of Eddy [30] and simulations of Margulies and coworkers [70], these genome sequences should provide resolution of selective constraint down to a segment length of eight nucleotides – approaching the same scale as individual transcription factor-binding sites. If successful, this strategy is likely to be applied to an even greater number of mammalian and other genomes (a fruit fly-based project is also currently under way; http://rana.lbl.gov/drosophila/multipleflies.html) the most exciting of which from the perspective of human biology is the proposal to sequence multiple primate genomes (http://www.genome.gov/12511814).

4.2 Defining and Obtaining Genomic Sequences

When undertaking a comparative genomic study, it is necessary to delineate a locus or loci of interest and to obtain corresponding homologous, often orthologous, sequences. Typically, an approximate locus will be defined by either arbitrary distances from an identified feature of interest, the confidence intervals of a preceding genetic study or the extent of a sequenced genomic fragment. It can be useful to extend a region of analysis slightly beyond the minimal extent so that the region is bounded by features that are well conserved between species, e.g. protein-coding exons, that serve as anchors for the analysis. A pair of well-conserved anchors provides confidence that the full extent of a locus has been isolated from each species under analysis.

Pre-assembled genomes are the most accessible source of defined genomic segments, as the problems of stitching together overlapping sequence fragments have already been tackled and the assemblies will have been subject to some degree of validation and quality control. Complete assemblies can be obtained from a number of disparate sites depending on the organism and assembly method. However, the UCSC Genome Browser, Ensembl (http://ensembl.org) and the National Center for Biotechnology Information (NCBI) (http://www.ncbi.nlm.nih.gov) all provide portals to the most current, and archived, public assemblies. These sites also provide means of searching the assemblies, such as BLAST [2], BLAT [56] and SSAHA [82], as well as precomputed annotation for the genome assemblies that can be readily incorporated into comparative genomic analyses.

There are several routes to identifying homologous loci in target genome sequences. An obvious approach is based on sequence similarity searches, but caution must be taken to distinguish orthologous from paralogous loci. Processed pseudogenes, in particular, are common [99], these are the reverse-transcribed copy of an mRNA that has integrated into the genome, but which does not code for a functional protein. As processed pseudogenes lack introns, they can score better than an orthologous locus in a similarity search. Genome-wide, reciprocal best matches [106] can be used to increase confidence that two loci are orthologous. Ensembl also provides precomputed assignments of gene orthology, currently based on reciprocal best matches for several genomes in the "geneview" pages and from the EnsMart data repository. Conservation of the order and orientation of genes in and neighboring the locus can also provide additional support of the orthology of two loci.

Probably the simplest currently available route to identifying orthologous loci is with the Net alignments at UCSC. These genome-to-genome pairwise alignments show genome-wide best matches and local rearrangements within them. They provide a direct means of jumping between an orthologous location in two genomes and can be used directly to delineate an orthologous locus in a target genome. For example, with the genome browser showing a complete locus of interest in a human assembly, clicking on the human to dog Net will provide an option to open the dog genome browser in a corresponding window from which the canine sequence and associated annotation can be obtained. An extension of this method is to use the genomic alignments to transfer annotation from one perhaps well-annotated genome to another that may have been recently assembled. The LiftOver tool at the UCSC Genome Browser (http://genome.ucsc.edu/cgi-bin/hgLiftOver) provides this facility for a limited set of genome pairs. This can provide a rapid way to get a baseline annotation which can then be filtered and refined. The Net alignments are generally good quality, but problems do arise, particularly where segmental duplications and assembly gaps are involved.

If there is uncertainty in the assignment of paralogy or orthology between multiple sequences, it can often be resolved through rigorous phylogenetic analysis (Chapter 4) of either whole genomic alignments or more discrete regions such as protein-coding sequences within them. This is often a problem with comparisons involving teleosts such as pufferfish and zebrafish, which may have been subject to a past whole-genome duplication [47] with the subsequent loss of many genes.

5 Technology

There are three general challenges that are common to most comparative genome analysis: (i) the production of an alignment, (ii) visualization of the alignment and (iii) detection of departures from neutral sequence evolution in the alignment. As alignments form the foundation of the comparative analysis, we will spend some time discussing the different options available and the consequences for interpreting results. There are also several options available for the visualization of large-scale genomic alignments. We have already discussed the principles and general approaches taken for the detection of departures from neutrality (Section 4); in Section 5.3, we present the tools that are currently available to apply these methods.

5.1 Alignments

The starting point for the majority of comparative genomic analyses is an alignment between homologous sequences. Precomputed alignments are available between several whole genomes as well as tools (Table 1) for producing such alignments. To a large extent, the genomic alignment tools and precomputed alignments can be treated as "black boxes". It is not necessary to understand in fine detail the process of producing the alignment to address a biological question with it. However, knowing in general terms how an alignment was generated, and the parameters used, can be crucial to its meaningful interpretation, especially when considering the apparent absence of conservation. Here, we present an overview of the genomic alignment problem, highlighting the limitations of available methods as well as recent advances in the field.

There are two general approaches to sequence alignment – local alignment and global alignment. Both of these strategies are introduced and discussed in detail in Chapter 3. When performing a *local alignment*, one is asking to be shown every similarity, scoring above a predefined threshold, between two sequences. The aligned subsequences (alignment segments) need not be in the same order or orientation in the parent sequences and one-to-many matches are permitted. In contrast, in a *global alignment* the entire length of one sequence is aligned with the entire length of the other through the insertion of gaps in both sequences. There is a maximum one-to-one correspondence between nucleotides, and their order is constrained such that duplications, inversions and other rearrangements cannot be detected. Rather than competing and redundant, these approaches should be considered complementary as they provide different insights into the relationship between two or more sequences.

Table 1 Summary of widely used and recommended genomic alignment tools

Program	Alignment method	Alignment tool	Ref.	Comment
AVID	global	pairwise	12	http://genome.lbl.gov/vista
BlastZ	local	pairwise	94, 95	the most widely used local genomic alignment tool; http://pipmaker.bx.psu.edu/pipmaker
Blat	local	pairwise	56	efficient use of memory and rapid execution make this a good choice for defining approximate regions to align with more sensitive methods
CHAOS	local	pairwise	15	by itself it lacks the heuristic refinements of BlastZ, but is used by DIALIGN and Lagan to identify initial alignment matches
DIALIGN	global	multiple	76	only practical for alignment of large (more than 10 000 nucleotides) sequences when used in conjunction with CHAOS [18]; http://dialign.gobics.de/anchor
GLASS	global	pairwise	6	one of the first available tools, now superseded by AVID
Lagan	global	pairwise	16	http://genome.lbl.gov/vista
MAVID	global	multiple	13	http://genome.lbl.gov/vista
mLagan	global	multiple	16	http://genome.lbl.gov/vista
MultiZ	global	multiple	10	based on BlastZ local alignments, but with a tiling path of aligning segments chosen (chaining, see main text) and integrated into multiple sequence alignments; this is the method used to produce high-resolution alignments for the UCSC Genome Browser (http://genome.ucsc.edu)
sLagan	global	pairwise	17	also known as Shuffle-Lagan – produces "glocal" alignments which have relaxed some constraints of global alignment so inversions, translocations and duplications can be detected; http://genome.lbl.gov/vista
TBA	global	multiple	10	a prototype standalone tool to produce threaded blockset multiple sequence alignment, similar to the output of MultiZ
WABA	global	pairwise	57	readily handles large gaps and can predict the protein coding/noncoding status of a sequence region based in part on the periodicity of divergent sequences

Note that several visualization tools, such as MultiPipMaker, emulate multiple alignment by stacking the percentage identity plots of multiple pairwise alignments without actually producing a character-based multiple alignment.

5.1.1 Local Genomic Alignments

All of the local alignment methods commonly applied to genomic sequences (Table 1) employ an index-based search strategy based on the same principle as that employed in the original BLAST algorithm [2]. Briefly, this approach produces an index of all *k*-length words (*k*-mers) in one of the input sequences and searches the other sequence for identical words. When a match is found, it is extended in both directions to define a maximally scoring segment of alignment. If that score is above the predefined threshold the alignment is reported.

There are three methods applied to genomic local alignment tools that elaborate this basic procedure to increase the sensitivity and specificity. (i) Requiring two matching words to be separated by a maximum distance from each other. This is a common approach used by BLASTN and most of the local alignment methods in Table 1. The principal exception is CHAOS, which identifies multiple matching words but does not perform alignment extension. Instead, the matching words are clustered (*chained*) if they lie in the same orientation and within a threshold distance of each other. It is this chain of words that is scored by CHAOS rather than BLAST-like extended initial matches. (ii) Using degenerate *k*-mers, which can tolerate a mismatch in any position of the *k*-mer, is a strategy that adds considerably to the computational load in the initial search step but provides more flexibility in defining word matches. This method is used by CHAOS in conjunction with the novel chain-of-words approach. (iii) Matching *k*-mers of nonconsecutive positions, an idea introduced to the field by Ma and coworkers [66]. For example a $k = 8$ word could be represented as 11111111 where each "1" denotes the position of an identity required for a match – a nonconsecutive $k = 8$ could be represented as 11011011011. This is distinct from the degenerate *k*-mer approach as a degenerate *k*-mer can tolerate a mismatch in any position, whereas the position of possible mismatches is constrained in the nonconsecutive *k*-mer case. Such patterns of matches can relate more directly to the underlying biology. The previous example could be useful to identify matches between coding sequence given the periodicity of codon conservation, due to the degeneracy of the genetic code. The nonconsecutive *k*-mer also has a slight statistical advantage over the consecutive *k*-mer, as the failure to match overlapping nonconsecutive *k*-mers is less strongly correlated between *k*-mers than the failure to match those which are overlapping and consecutive [5].

Beyond the limits of sensitivity defined by the initial index search, there are many parameters that can be modified in the available tools to optimize them for a specific purpose or phylogenetic scope. For such insight we direct the reader to the primary literature and associated web servers (Table 1). However, the program BlastZ is one of the most versatile and widely used in this class of program for comparative genomic studies and is the basis for a number of publicly available resources, as such we consider its use in more detail here.

Developed principally by Scott Schwartz and Webb Miller [94, 95], BlastZ is based on the Gapped-BLAST algorithm [2]. An alignment is seeded by a short perfect or defined imperfect match, extended by dynamic programming, initially without gaps and if score thresholds are achieved, then with gaps. Sequence between anchoring alignments is again searched and alignments extended, but using a lower stringency than in the initial search, the stringency being determined by the separation distance between anchors. BlastZ employs heuristics to take sequence complexity into account, requiring low-complexity sequence to align better than high-complexity sequence and to dynamically mask any regions with an unexpectedly large number of matches. As BlastZ is a local alignment tool, matches may overlap, they can be distributed between both strands and are unconstrained in their linear order. However, BlastZ has the option of constraining matches to be colinear between input sequences (chaining) or to select only a best match to each region of a reference sequence (single coverage). Both of these options involve discarding data, but can be useful in interpreting results and subsequent analysis.

5.1.2 Global Genomic Alignments

The prototypical global alignment method is that of Needleman and Wunsch [78]. However, this procedure does not scale well to the large alignments commonly required in comparative genomics. The approach employed by most of the genomic global alignment tools is to define a series of *anchors* – high-confidence matches between a pair of sequences that are constrained to be in the same order and orientation in both sequences. This is effectively the chaining method optionally employed by BlastZ, as discussed above. The portion of each sequence between adjacent anchors is then aligned with lower stringency, defining a new set of anchors and the process is reiterated until all sequence is aligned. The strategy effectively breaks the large alignment down into a series of progressively smaller alignments, with two important consequences. (i) The total search space is quickly reduced and continues to be refined with each iteration, allowing the alignments to be produced quickly and using little memory relative to the length of input sequences. (ii) The chain-of-anchors approach is tolerant of large gaps, which are common in genomic sequence alignments, but poorly dealt with by gap penalties employed by the purely dynamic programming methods such as Needleman–Wunsch (Chapter 3).

Table 1 summarizes the global sequence alignment tools that are often applied to genomic sequences. Of these, AVID [12] and Lagan [16] are the most widely used. AVID identifies maximal matches (identical runs of nucleotides) and from these selects a chain of nonoverlapping alignment anchors using dynamic programming, iterating the process as described above until all bases are aligned or there are no significant matches in the remaining subsequences. The full Needleman–Wunsch algorithm is applied if the remaining sequences are short (less than 4 kb), otherwise a fully gapped alignment of these regions is returned.

A particularly useful feature of AVID is its ability to perform template-directed fragment assembly. Provided with a contiguous and a fragmented sequence, AVID will use high-confidence local matches to order and orient the fragmented sequence relative to the contiguous one, producing a "merged draft" which is then used for pairwise alignment. Such a utility can be invaluable in the analysis of early-draft genomic sequences.

Lagan [16] proceeds in a very similar iterative manner to that described for AVID, making use of the application CHAOS to produce local alignments from which the anchors are defined. In a further development, Brudno and coworkers [17] have generalized this approach by relaxing the criteria for colinearity in the order of alignment anchors, instead requiring them to be sequentially ordered along only one of the input sequences, the designated reference sequence. This relaxation allows for the detection of genomic rearrangements such as inversions, translocations and duplications relative to the reference sequence. This method is implemented as Shuffle-Lagan (Table 1). In recognition of similarities to both local and global methods, the authors have termed these "glocal" alignments. The approach is innovative and has potential to be developed further, but there are two key drawbacks to the current implementation: (i) as the two input sequences are treated differently, the resulting alignment depends on the order sequences are presented, and (ii) our current lack of understanding of the frequency of genomic rearrangements to appropriately parameterize such alignments.

5.1.3 **Multiple Sequence Alignments**

The local and global sequence alignment methods we have discussed so far are only able to produce pairwise alignments. We have seen in Section 3, however, that the combined analysis of multiple sequences provides much greater insight, statistical power and resolution to comparative genomic studies. Unfortunately, the difficulties of producing pairwise genomic sequence alignments are exacerbated in the challenge of producing multiple alignments.

To perform a progressive multiple alignment in this manner, the phylogenetic relationship between sequences being aligned needs to be established. This can either be calculated from initial all-versus-all pairwise alignments of the sequences or for some programs can be provided in the form of a previously established phylogenetic tree. If the multiple sequence alignment is between orthologous sequences, their relationship is often known in advance, e.g. (((human,chimp),(mouse,rat))dog). Provision of the tree in advance removes uncertainty in the order a program aligns the sequences, providing consistency between the alignment of multiple loci and expediting the alignment process, as well as ensuring the correct phylogeny is used. After producing the initial multiple sequence alignment, the location of gaps can be optimized, making use of the greater information content of multiple sequences. There are multiple alignment versions of AVID and Lagan, denoted by the "M" prefix to the name (MAVID and mLagan) – both of which use the general method outlined above to produce the global multiple sequence alignments.

Although MAVID and mLagan produce true multiple sequence alignments, many visualization tools (Section 5.2) display conservation profiles relative to a chosen reference sequence. A similar approach is frequently used to integrate conservation measures between multiple pairwise alignments [94] and to define multiply conserved sequences (MCS) (Section 5.3). The reference sequence approach is an obvious choice if the objective is the annotation or investigation of a particular sequence. However, it is increasingly the case that comparative genomic studies intend to measure how a locus has evolved in multiple lineages and how selective forces have changed during that evolution, rather than just detecting regions of the reference sequence that are selectively constrained. For these analyses the reference sequence approach has two major drawbacks. First, any regions conserved between a subset of aligned sequences, but not the reference, will not be detected. This problem can be overcome by generating several multiple sequence alignments – one with each of the sequences under study as the reference. This solution is time consuming, raises the additional problem of integrating results between alignments and exposes the second major drawback to the reference sequence approach, i.e. the potential for inconsistencies when using alternate sequences as the reference.

A solution to the problems presented by reference sequence based alignment and analysis has been proposed in the form of a "threaded blockset" [10]. Under this proposition, a multiple sequence alignment is represented as a series of alignment blocks, termed the *blockset*. Within an individual block, each row corresponds exactly to an input sequence (or its reverse complement) if gap characters are ignored, i.e. no sequence within a block has been rearranged. Additionally, an individual block need not involve every aligned sequence. From this blockset, multiple sequence alignments can be produced using any one of the aligned sequences as the reference sequence, simply by ordering and orienting the blocks according to the selected reference sequence, a process referred to as threading the blockset. This approach ensures consistency of alignment when alternate reference sequences are used and no portions of the alignment are discarded. The threaded blockset aligner (TBA [10]) has been developed as a prototype tool to generate blocksets.

Many eukaryotic genomes are rich in repetitive sequences; these can confuse alignment programs if not treated appropriately. The identification of repetitive sequences is considered in detail in Chapter 7. The simplest treatment of interspersed repeats and low-complexity regions is to mask the sequence prior to alignment, readily achieved with tools such as RepeatMasker (A. Smit and P. Green, http://www.repeatmasker.org/) or available precomputed from the UCSC Genome Browser for a wide range of whole-genome assemblies. However, interspersed repeats can be interesting in their own right and are a useful measure of background mutation rate (Section 5.3). A more satisfactory treatment of repeats is to ignore them in the initial stages of alignment and then align through them if flanking nonrepeat sequence has been aligned. This is often termed "soft-masking", and is implemented in several alignment tools including LAGAN, Blat and AVID.

5.1.4 Assessing the Quality of Genomic Alignment Tools

Which alignment tool is the most accurate? This is an obvious question to ask when deciding which tool is the most appropriate to use. Unfortunately, this appears to be impossible to definitively answer. For protein-coding sequences, solved three-dimensional protein structures provide a "gold standard" against which alignment methods can be scored [14]. No such equivalent exists for noncoding DNA. A possible solution is the *in silico* simulation of sequence divergence [103], which can provide a population of sequences related to a common ancestor by a precisely known sequence of mutational events, so the true alignment is known.

There is a chance that an evaluation of alignment success based on simulated data is measuring the similarity of evolutionary models rather than the sensitivity and specificity of the alignment methods themselves. Despite this limitation, Pollard and coworkers [88] have performed such an analysis and produced some useful rules of thumb for genomic sequence alignment. All methods rapidly lost sensitivity with increasing divergence, with more than 50% of nucleotides not accurately aligned by all methods with $D = 1.0$ (divergence, substitutions per site) in the most realistic simulations. Local aligners were successful at identifying constrained sites, but performed poorly on neutral sequence with $D > 1.0$. As would be expected from their mode of action, global aligners had the highest overall sensitivity to accurately align orthologous sites in both neutral and selectively constrained sequence. Lagan [16] performed particularly well under almost all of the simulation scenarios. The simulations in this study did not include inversions and duplications which would have only been detected by the local alignment methods.

5.1.5 Using Whole-genome Alignments

As we have seen, there is a good diversity of tools available to produce pairwise and multiple genomic sequence alignments. Although these tools are optimized for genomic sequence alignment, the alignment of whole eukaryotic genomes to each other is still a daunting and specialist task requiring considerable computational resources. Fortunately, several research groups that specialize in such large-scale genomic alignments have made their alignments publicly available (Table 2). The utility of these alignments is not limited to whole-genome analyses and they represent an excellent resource for investigations focused on defined loci.

Table 2 Precomputed whole eukaryotic genome alignment resources

Resource	URL	Reference
MultiZ at UCSC	http://genome.ucsc.edu/	10
Berkeley Genome Pipeline	http://pipeline.lbl.gov/	25
GALA	http://gala.cse.psu.edu/	33, 39

These publicly available alignments have several significant advantages over proprietary alignments produced *ad hoc* to address specific questions. (i) As they are a public resource, they are used by many members of the research community to address a multitude of questions and, therefore, any systematic problems in their construction are likely to be highlighted, whereas in-house alignments are unlikely to be as rigorously vetted. (ii) Results based on the same alignments can be directly compared between research groups, e.g. in the integration of findings in large collaborative projects [40, 114]. (iii) It is faster and simpler than producing your own alignments, especially in the cases where existing annotation has already been mapped to the alignments (http://pipeline.lbl.gov) or can be readily mapped using easily accessible tools (http://pipmaker.bx.psu.edu/piphelper).

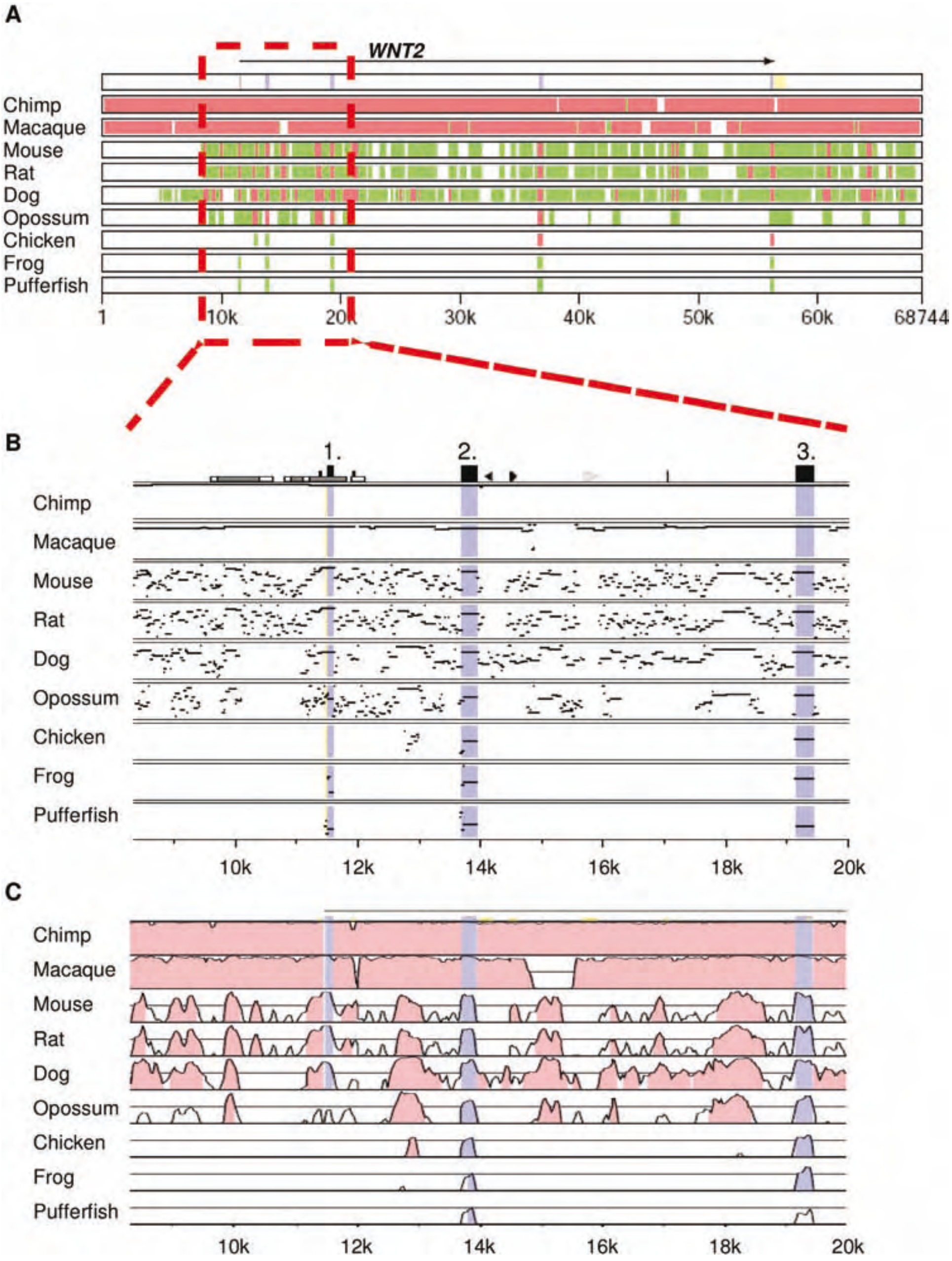
However, there are of course limitations to the utility of precomputed alignments. The user is restricted by the predefined phylogenetic scope of the alignments, e.g. at the time of writing, the human-based MultiZ alignments available from UCSC included alignments with chimp, mouse, rat, dog, chicken, pufferfish and zebrafish, but assemblies for the genomes of opossum, rhesus macaque, cow and frog are also publicly available, and could add considerably to the information content of the multiple sequence alignment. It is also the case that for some analyses very specific sets of alignment parameters or constraints are required [55], which are unlikely to be met by off-the-shelf whole-genome alignments.

5.2 Visualizing Genomic Alignments

The visual representation of alignment based data is an important aspect of comparative genomics, especially when the focus of the analysis is a locus of specific interest (see also Chapter 43). One of the most intuitive and logical representations of a pairwise sequence alignment is a dot plot (Chapter 3). Such a representation can summarize all regions of local similarity between two sequences, highlighting inversions, translocations, duplications and deletions. Plotting a sequence against itself is often an excellent first step in the comparative characterization of a locus as it can highlight regions that are tandem repetitive, low complexity and clearly show segmental duplications – all of which are potentially confusing to interpret when visualized with the other methods discussed below. For sequences of up to a few hundred kilobases, the Dotter software [102] is able to produce a complete dot plot and incorporate arbitrary annotation. For sequences above this size, the computation of a complete dot plot is impractical, but tools such as PipMaker [95] can produce dot plot-style summaries of local alignments (Figure 2) which can be interpreted in essentially the same way.

The downside to dot plots is that they take up considerable space and are impractical when it comes to summarizing the similarity between multiple sequences. For these reasons, percentage identity plots (PIPs) were introduced [44] in which the x-axis represents the coordinates of a reference sequence and the y-axis shows percent identity (Figure 2). A horizontal bar within the plot then identifies a gap-free segment of local alignment, the horizontal position and extent of the bar defining the aligning section of the reference sequence. The position of the bar in the y-axis shows the percentage nucleotide identity for the ungapped local alignment. This is a versatile way of displaying pairwise sequence similarity as it can be applied to both local and global alignments, and through stacking of multiple such plots can be adapted to show the conservation of a reference sequence aligned with any number of sequences.

Another intuitive and commonly used representation of nucleotide identity in sequence alignments is to plot a histogram of conservation (Figure 2). As with PIPs, identity is plotted against the coordinates of a chosen reference sequence. Rather than calculating the identity from an ungapped segment of alignment, however, it is calculated from a predefined range of nucleotides in the reference sequence. These can be discrete consecutive bins of, say, 10 alignment columns or, more commonly, calculated as a sliding window, e.g. VISTA [73] uses a window of 100 columns with sliding increments of 1, by default.


A
WNT2
Chimp
Macaque
Mouse
Rat
Dog
Opossum
Chicken
Frog
Pufferfish
1
10k
20k
30k
40k
50k
60k
68744
B
1.
2.
3.
Chimp
Macaque
Mouse
Rat
Dog
Opossum
Chicken
Frog
Pufferfish
10k
12k
14k
16k
18k
20k
C
Chimp
Macaque
Mouse
Rat
Dog
Opossum
Chicken
Frog
Pufferfish
10k
12k
14k
16k
18k
20k

5.3 Detecting Selection

Any significant departure form the neutral rate of sequence evolution can indicate the action of selection. The neutral rate is typically calculated from regions of alignment corresponding to one of four types of sequence: 4-fold degenerate (4D) sites, ancient repeats, anonymous sequence or pseudogenes. We consider the advantages and drawbacks of each of these below.

In the standard genetic code, there are eight instances where substituting the third codon position for any other nucleotide will not change the encoded amino acid (CTn = Leu, GTn = Val, TCn = Ser, CCn = Pro, ACn = Thr, GCn = Ala, CGn = Arg, GGn = Gly) – these are 4D sites. 4D sites are readily identified from annotated or well-predicted coding sequences and because they are embedded in generally well-conserved coding sequence they can often be aligned between even highly divergent sequences with a high degree of confidence. For these reasons, 4D sites represent an excellent type of sequence from which to calculate the branch length D. In general such sites are readily identified as less conserved than other coding positions and non-4D third codon positions [79]. However, that is not to say that they are devoid of function or functional constraint – such sites may be involved in the regulation of splicing, translational efficiency, mRNA localization or stability. 4D sites are generally considered to be good for the calibration of nucleotide substitution rates, but they are of no use in measuring the neutral rate of insertion, deletion or rearrangements.

←

Figure 2 Visualization of genomic sequence alignments. The WNT2 locus was aligned between human and orthologous loci from nine other vertebrates for which at least a draft whole-genome shotgun sequence is available. Orthologous regions and extents were defined using the UCSC Nets. In each case, coordinates and annotation are shown for the human sequence and nucleotide identity from pairwise alignment. (A.) Summary view from MultiPipMaker [94] based on BlastZ alignments. The extent of the WNT2 transcript is shown above the alignment, protein-coding exons indicated in purple and untranslated region (UTR) in yellow. Regions of local alignment are shown in green or red if a combined length and identity threshold is achieved. The region highlighted is shown in detail in (B) and (C). (B.) Detailed view of MultiPipMaker output – a percentage identity plot. Exons are denoted by black boxes above the plot and projected as purple-shaded regions across it. Other features above the plot correspond to annotated repetitive elements (triangles and predicted CpG islands (grey and white boxes). (C.) VISTA plot [73] summarizing mLagan [16] global alignments of the sequences. Higher curves show greater conservation; regions meeting a threshold level of conservation are shaded – purple for sequence annotated as protein coding, blue for UTR and red for anonymous sequence. Exons 2 and 3 are readily aligned in all cases, whereas the relatively short and poorly conserved exon 1 is not always aligned (panel C, frog and pufferfish). An additional complication when using draft sequences is the presence of assembly gaps – the apparent failure to detect exon 1 in chicken in this case coincides with a gap in the chicken assembly.

IREs are widespread through most vertebrate genomes and are thought to be free from selective constraint (Section 2). Unlike 4D sites, IREs are free to accumulate insertion, deletion and rearrangement as well as substitution changes [85]. Using the tool RepeatMasker (http://repeatmasker.org) combined with an appropriate repeat database (Repbase; http://girinst.org) IREs can be grouped into families and subfamilies based on sequence similarity. Each copy of an IRE subfamily is thought to have been almost identical at the time of insertion as they were all produced from one or a very small number of "parent" elements in a brief period of activity before mutation robbed the parent element of its ability to transpose [59]. Therefore, an IRE that inserted into a genomic location in the common ancestor of a set of sequences being compared is expected to accumulate mutational changes independently in each of the diverging lineages and those changes are likely to be invisible to selection.

This assumption of identity between IRE subfamily members at the time of insertion provides them with additional advantages over other categories of candidate neutral sequence. For example, using the IRE subfamily consensus sequence as an out-group, mutational changes can be assigned both a direction and a lineage from just pairwise comparisons rather than requiring a minimum three aligned sequences.

For mammalian and other genomes rich in interspersed repeats, IREs appear to be the ideal solution to measuring the neutral rate of mutation. However IREs are typically defined on the basis of their sequence similarity to previously defined repeats and to other sequences in the genome. This means that highly diverged members of a repeat family may not be detected, resulting in an under estimation of the mutation rate. The distribution of IREs is nonrandom across a genome [45], some favoring A/T-rich insertion sites, others showing preferential retention based on nucleotide composition. The nonrandom distribution may result a systematic bias in mutation rate estimation. The abundance of these elements in the genome may also lead to nonorthologous recombination between elements [53], resulting in a high frequency of gene conversion within the elements [91].

Another possibility is anonymous sequence. In genomes dominated by nonfunctional sequence such as those of mammals (Section 2), the background mutation rate can be approximated by simply taking the average rate across the whole alignment. This estimate can be improved by specifically excluding annotated functional sites such as protein-coding exons and core promoters. The remaining unannotated (anonymous) regions of alignment will be enriched for selectively neutral sites. An interesting variation is to use sequences that align between closely related species but do not align with a more distant out-group, because the sequence has been inserted in one lineage or lost from the other [24]. Again, it can be argued that the sequence is less likely to contain important functional elements and is thus enriched for selectively neutral sites.

Pseudogenes (Section 2) are particularly interesting for estimation of the neutral rate because their starting point is a functional gene, with all the associated sequence biases, periodicity, and, in the case of nonprocessed pseudogenes, introns, splice junctions and regulatory sites. These are often the features we are most interested in identifying or investigating in comparative studies. If a gene pseudogenized before the common ancestor of compared sequences, we can see the effect of mutation and genetic drift free from the action of selection superimposed upon it. This is the ideal scenario. Unfortunately nonprocessed pseudogenes are too rare – only 37 having been found in a systematic screen of the human genome [59] – to be of general use in calculating background mutation rates. Processed pseudogenes have been useful for the investigation of protein-coding sequences [84], but again their uneven distribution limits their use in estimating local mutation rates and some sequences identified as pseudogenes may still have functional roles [87].

For a comparative study that aims to identify highly constrained sequences that are evolving many times slower than the neutral rate, it can be adequate to estimate the neutral rate on a genome-wide basis. Such studies include the identification of protein-coding sequence with distant pairs of sequences, e.g. human versus pufferfish [26, 107], and the ultra-conserved elements [7] discussed later (Section 6.2). For more sensitive studies it is necessary to calculate the neutral rate in localized regions, as the rate of mutation has been found to vary spatially across genomes [45, 117] (Section 2).

Regional estimates of the neutral rate can be calculated in a sliding window manner or by calculating it for an arbitrarily defined region of interest. The principal problem with this approach is that sites subject to selection cannot be assumed to be randomly distributed across the genome. For instance, anonymous sequence around the *PAX6* gene [74] is highly enriched in functionally important conserved sites (Section 6.3). An estimate of the neutral rate based on anonymous sequence around this gene would give an artificially low estimate for the neutral mutation rate in the region. The larger the window used to estimate the regional neutral rate, the less likely it is to be dominated by non-neutral sites, but a larger window reduces the resolution for detecting regional variation in mutation rate. The optimum window size for neutral rate estimation will be a balance of these two opposing needs. If recent findings [38] for the rodent lineage can be generalized to other vertebrates, then typically a window of 10 kb is likely to show a consistent neutral rate across its length and even windows up to 1 Mb may have little variation in neutral rate across them, but substantially larger than this and neutral rate fluctuations may be compromised by neutral rate variation. Testing for concordance between different measures of the neutral rate such as those derived from 4D, ancient repeat and anonymous sequence allows the influence of cryptic non-neutral sites to be evaluated and provide a justification for the selected window size.

Having established a neutral rate estimate for the sequences of interest, the challenge is to identify and delimit regions of sequence that significantly deviate from it. At its simplest this can be a relative rate test, dividing the rate of evolution in the test sequence by the neutral rate. Values of one indicate neutrality, significantly less than one, constraint and more than one indicates diversifying selection. There are many variations on this basic principle, including the commonly used K_a/K_s ratio test, where K_a denotes the rate of substitutions that lead to amino acid changes and K_s the rate of substitutions in protein-coding sequence that do not lead to a change in the encoded amino acid. K_a/K_s equal to 1.0 indicates neutral evolution (no selection), below 1.0 indicates purifying selection and above 1.0 positive (diversifying selection) to detect selection in the protein-coding sequence [48]. The differences between approaches predominantly relate to how to score constraint between multiply aligned sequences and how to define the boundaries of a constrained sequence.

Several studies have defined the extent of constrained regions on the basis of ungapped segments of alignment [28, 29] – a strategy that lends itself well to analyses based on local rather than global alignments. Often, these studies will use precalibrated thresholds for significant constraint rather than calculating relative rates directly, e.g. 70% identity over 100 ungapped nucleotides are commonly used parameters for human to rodent alignments [28].

Sliding windows have been widely used to arbitrarily define the extent of sequences which are then evaluated for constraint [73, 114]. The approach can accommodate alignment gaps, generally treating them as nucleotide mismatches [73], but their sensitivity is crucially dependent on the size of the evaluation window and by how much the window is moved along the alignment for each evaluation. Analyses based on sliding windows have also been applied to phylogenetic shadowing [11]. In this case, rather than scoring conservation or substitution rate directly, the substitution rates for each alignment column were compared to the rates of sequences known to be evolving neutrally or subject to selection. The final score being a likelihood ratio of neutral versus constrained evolution for each alignment column. A web server for phylogenetic shadowing analysis is available (http://bonaire.lbl.gov/shadower).

An intuitive way of integrating measures of constraint across multiple aligned sequences is to define MCS – the common core of sequence that aligns in all (or most) sequences from a defined scope can then define the boundaries of the MCS [69, 110]. For instance it is easy to see that exons 2 and 3 of *WNT2* can be considered MCS within vertebrates (Figure 2). The MCS definition is versatile, accommodating local or global alignments and can tolerate missing sequence from incompletely sequenced genomes.

Two highly versatile tools, RankVISTA [71] and phastCons [100], have recently been developed that quantify constraint and operate free of window size constraints and identity thresholds. These tools are also noteworthy because they quantitatively measure constraint rather than the crude binary discrimination into constrained or unconstrained that is common to many of the methods discussed above. RankVISTA integrates pairwise relative rate scores across a multiple alignment using a phylogenetic weighting scheme. The neutral rate estimates are derived from anonymous regions in the submitted alignment and the final score is an easily interpretable probability of observing such conservation in a 10-kb fragment of neutrally evolving sequence. The extent of constrained sequences is determined with dynamic programming (Chapter 3). This tool is available from the standard VISTA web server (http://genome.lbl.gov/vista). The tool phastCons [100] is one of the first practical implementations of a phylogenetic hidden Markov model (phylo-HMM [36]) to score conservation across genomic alignments, in effect scoring how well the observed pattern of substitution matches its internal model of a constrained site. The approach is also noteworthy because it takes into account the tendency for conservation levels to be similar at adjacent sites and it is an extensible model that could incorporate additional parameters. Precomputed phastCons results are available through the UCSC genome browser for multi-species whole-genome alignments.

All of the methods described above focus on nucleotide substitution rates. Insertions and deletions also have the potential to help detect constrained regions; however, estimation of their rate is more sensitive to alignment parameters than is the case of substitution rate calculations [54], and good stochastic models of insertion and deletion in noncoding DNA are not currently available. Alignment gaps are typically treated as either missing data (phastCons) or nucleotide substitutions (RankVISTA) when assessing selective constraint. Neither of these are particularly satisfactory solutions and in the case of phastCons lead to artificially high scores over regions of gapped alignment. A better treatment of alignment gaps is likely to be an important avenue for future work in the development of these methods.

6 Applications

There have been a huge number of published studies that are either centered on comparative genomic analysis or utilize comparative genomics to address specific questions within a wider study. In the next few subsections we highlight a small number of examples that have given new insight into the general biology of vertebrate genomes and provide good examples of the application of methods described in this chapter.

6.1 How Much of the Human Genome is Constrained?

The protein-centric view of genome function (Section 2) has been challenged by recent whole-genome comparative analyses. With publication of both the draft human [59] and mouse [114] genomes it became possible to estimate the total proportion of the human genome that is subject to selective constraint, and so estimate the proportion of the genome that has conserved function, but which is not protein coding. A conservation score was calculated for nonoverlapping 50 nucleotide windows of human:mouse whole-genome pairwise alignments. Two sets of scores were derived, one for the complete alignment and a second derived only from aligned ancient repeats. As ancient repeats are thought to be unconstrained by selection (Section 5.3), the distribution of conservation scores should reflect the pattern expected under neutral evolution. The distribution of scores from the whole-genome alignments substantially overlaps those derived just from ancient repeats, although a significant shoulder specifically towards higher scores is evident [114]. Subtracting the ancient repeat distribution from that of the whole genome suggests that approximately 5% of 50 nucleotide windows are more highly constrained than expected under neutral evolution. Similar analyses based on human to rat comparisons have supported this conclusion [40]. These studies are not without their limitations. In particular, isolated regions of constraint that are substantially shorter than the 50-nucleotide window size used will have gone undetected, suggesting that the estimate of 5% may be a lower bound for the true value. Using a novel method based on insertion and deletion rates rather than substitutions also finds that generally similar fractions of 2.6–3.5% of the human genome show evidence of selective constraint [64]. These studies have led to the important conclusion that much of the functionally constrained sequence in the human genome does not code for proteins.

If coding sequences are not the singularly dominant functional component of the genome, the question arises “What are the functions of noncoding sequence?”. Several types of noncoding elements are known, such as *cis*-regulators of transcription and splicing (see Chapter 6), RNA structures that influence transcript localization and stability, as well as transcripts whose functional product is RNA rather than protein (see Ref. [72] for review). It is also likely that there are classes of functional elements that we have yet to discover. This potential naivety is well illustrated by the relatively recent realization that a major class of noncoding functional elements (microRNAs) had been almost entirety overlooked [3].

It is one of the great strengths of comparative genomics that no prior assumption of the function is required to identify a sequence as functionally important. With the increasing depth of available genomes (Section 5) and the methods described above, we are rapidly approaching the stage where we can confidently identify short regions and possibly even single nucleotides as constrained. A remaining and significant challenge is to characterize the function of those sites. Again, comparative genomics can help. We have already seen that there is a characteristic profile of conservation for protein-coding sequence (Section 5.1) and similar profiles may exist for other categories of functionally important sequence. Dermitzakis and coworkers [27] found that conserved nongenic sequences (CNGs) accumulated sequence changes in a manner that can be statistically distinguished from both protein-coding sequences and noncoding RNA genes. These patterns of sequence change most resembled clusters of protein binding sites.

6.2 Ultra-conserved Regions

The sequences studied by Dermitzakis and coworkers [27] were selected, from chromosome 21, on the basis of a simple threshold identity in human to mouse alignment, and also on the ability to polymerase chain reaction amplify homologous sequences from 14 mammalian species. Consequently, these sequences should represent the subset of CNGs that have both the highest nucleotide identity and are the most constrained through mammalian evolution. Ironically, a whole-genome analysis of noncoding conservation has since shown that human chromosome 21 is the only autosome devoid of so-called ultra-conserved elements [7]. These elements are also defined on the basis of simple length and identity thresholds, but in this case 200 nucleotides of ungapped alignment between human, mouse and rat with 100% nucleotide identity in all three species. In total, 481 of these incredibly well conserved sequences were found.

Although defined initially on the basis of conservation between humans and rodents, 97% of the ultra-conserved elements could be identified in the chicken genome with, on average, more than 95% nucleotide identity and more than 66% of them could be aligned with a pufferfish genome (*T. rubripes*). In contrast, only 5% could be identified in any of the nonvertebrates *Ciona intestinalis* (sea squirt), *Drosophila melanogaster* (fruit fly) or *Caenorhabditis elegans* (nematode worm) and all of these were ultra-conserved elements that overlap protein-coding exons from known genes. It appears then, that although these ultra-conserved elements have been highly constrained for 300–450 million years of vertebrate evolution [7], they are largely confined to the vertebrates. A similar study making use of recently available whole-genome sequence from multiple insects, has also identified ultra-conserved regions between fruit flies and the mosquito *Anopheles gambiae* [42]. However, the majority of ultra-conserved elements identified in files were substantially shorter than the 200 nucleotide threshold used for the human study despite similar evolutionary distances for some of the analyses in both studies.

It has been noted in both mammals and flies that ultra-conserved elements are often located in the introns of, or intergenic regions around, developmentally important genes [7, 42, 118]. These developmental regulators are often DNA-binding transcription factors or RNA-binding proteins [7] that are likely to be involved in the regulation of RNA processing and transport. These observations have invoked the notion of developmental master regulators – regions that integrate multiple signals coordinating the expression of genes which in turn regulate many more genes through transcription and RNA processing. Direct experimental support for this idea has been provided by Woolfe and coworkers [118] in a zebrafish experimental system. Of 25 noncoding sequence elements that are highly conserved between human and pufferfish, 23 showed significant transcriptional enhancer activity in one or more tissues during zebrafish development [118].

The idea that ultra-conserved elements act as developmental regulators fits well with the observation that they are highly conserved within phylogenetic clades that share similar developmental programs, but apparently not conserved between more diverse groups. Could the ultra-conserved elements that are common to both humans and pufferfish be the master regulators that define the basic vertebrate body plan: skeletal structure, musculature, internal organs and the developmental programs to orchestrate their construction? This is an attractive idea, but much more work is required to establish if this is even close to accurate. In particular, some genes are known to be key regulators of developmental programs, and the orthologous genes in both humans and flies are apparently performing the same task in the same tissue. *PAX6*, for example, is crucial in the development of eyes in both human and fruit flies [111]. The human *PAX6* locus is one of the richest in ultra-conserved elements [7] and six out of seven tested elements show enhancer activity, four of which directed expression preferentially in the developing eye [118]. Despite the conserved role of *PAX6* in eye development between humans and flies, and the demonstrated role of mammalian ultra-conserved elements in directing that expression, there is no identifiable sequence similarity between the ultra-conserved elements and the fruit fly *PAX6* locus.

6.3 Specific Locus Studies

In this section we focus on a small number of disease related studies that have been substantially advanced through the application of comparative genomics. We make several references to Online Mendelian Inheritance in

Man (OMIM) – a key human curated resource that brings together published information relating human genetic diseases and disease genes. Full OMIM records can be obtained with their identifier number from the Entrez system (http://www.ncbi.nlm.nih.gov/entrez).

Hirschsprung disease is a congenital disorder characterized by intestinal abnormalities (OMIM:142623). The genetics of this disease have been well studied, but the pattern of inheritance is complex. Mutations have been found in eight loci which contribute to disease susceptibility (OMIM:142623 for review), but these account for only 30% of cases [34]. Genetic evidence indicated that one of those eight loci, the *RET* gene, harbored additional, previously undetected mutations or variants that account for much of the remaining susceptibility [37]. All apparent protein-coding sequence of *RET* had already been screened for mutations, so the challenge was to identify additional functionally important noncoding sites within the locus or identify previously missed protein-coding sequence.

Emison and coworkers [34] identified more than 30 regions of conserved noncoding sequence in 350 kb of genomic sequence centered on the *RET* gene. The analysis used the multiple conserved sequences paradigm (Section 5.3) based on alignment of 12 orthologous loci from vertebrates. Only five of the conserved noncoding regions were within the region maximally implicated by genetic evidence. The comparative analysis also indicated that a human single nucleotide polymorphism (SNP) is located within one of the conserved regions and, not withstanding the polymorphism, the nucleotide has been highly conserved through vertebrate evolution. An obvious candidate for the functional variant, Emison and coworkers [34] were able to show that this conserved element has enhancer activity and that the level of that activity is influenced by the SNP genotype. The comparative alignment allowed the ancestral and derived alleles to be discriminated – the lower enhancer activity and disease susceptibility being associated with the more recently derived allele. The noncoding SNP genotype was shown to account for much of the previously unaccounted for genetic susceptibility contributed by the *RET* locus.

The Hirschprung disease *RET* locus is a good recent example of the utility of comparative genomics and its synergy with genetic studies. It also stands out for several of other reasons. The functional variant identified is common in the population, exceeding 50% in some parts of East Asia, despite being disease-associated. The effect of the genotype is influenced by sex, demonstrating a form of epistasis, and the variant is regulatory rather than protein coding. All of these features are likely to be frequently encountered when searching for the genetic risk factors in common diseases [68] such as cancer, heart disease, diabetes and stroke.

The *RPGR* gene has a similar story to the *RET* locus. *RPGR* is known to be a major locus for X-linked retinitis pigmentosa (OMIM:312610) – a form of retinal degeneration. Several known disease associated coding sequence mutations had been found, but it was apparent from genetic studies that many more cases of retinitis pigmentosa should be attributable to the locus than could be explained by the mutations in the coding sequence [109, 113]. Comparative genomics revealed a previously unknown, alternately spliced protein-coding exon that was specifically expressed in the retina and harbored the missing mutations [113]. In this case all of the disease-associated mutations disrupted the encoded protein. It is likely that such missing mutations are common for even well-studied genes and that they are simply under-reported in the literature, because it is seldom practical to screen large genomic intervals for mutations, nor is it easy to demonstrate their causal role. Our next example demonstrates over how wide an interval *cis*-regulatory sites can act, but also how, even when the region is large and complex, comparative genomics can allow functional sites to be identified an subsequently characterized.

The mouse *Sasquatch* (*Ssq*) mutation was generated serendipitously when trying to insert a transgene into the genome. The transgene integration led to ectopic expression of the developmental signaling molecule Sonic Hedgehog (SHH) and resulted in preaxial polydactyly (extra digits) [97]. Intriguingly, genetic evidence demonstrated that the effect was specifically in *cis* [97], but as the integration site was over 1 Mb from *Shh* and located within the intron of an adjacent gene, identifying the functional regulatory element remained a challenge.

Multiple sequence alignment between orthologous regions from mouse, human, chicken and pufferfish identified a 0.8-kb stretch of sequence close to the transgene insertion site that had been highly conserved throughout vertebrate evolution [61]. It has now been shown that the 0.8-kb element, known as the ZRS, is a limb bud-specific enhancer of *Ssh* expression [61, 92] and that even the fish sequence can drive expression in the mouse limb bud. These studies into the *Shh* locus have shown that *cis*-regulatory elements can be located large distances, at least 1 Mb, along linear DNA from the genes they act to regulate. Not only can these elements be far from their targets, they may also be closer to other genes on which they apparently have no regulatory role – the ZRS is located in the fifth intron of the *Lmbr1* gene whose expression is unaffected by mutations in the ZRS.

Like the ultra-conserved sequences described above, the striking conservation of the ZRS through vertebrate development indicate that even single nucleotide substitutions in the region are likely to be detrimental and strongly selected against. Accordingly, point mutations in the ZRS have been found in four human families and two mouse lines, in each case they lead to preaxial polydactyly (see Ref. [62] for review). In contrast to these point mutations, complete deletion of the ZRS in the mouse abolishes *Shh* expression in the limb bud and results in severely truncated limbs [92] – a similar phenotype to human acheiropodia which is also linked to the *Shh* locus [49]. Several vertebrate lineages such as snakes have substantially reduced or entirely lost limbs, although they were present in their ancestors. Sagai and coworkers [93] have shown that for at least two of these cases, i.e. snakes and limbless newts, this has coincided with the loss of the ZRS, where as it remains conserved in lizards and legged newts. Whether loss of the ZRS was a primary event in the morphological transition of either of these separate lineages or if it represents secondary losses remains unclear; however, it does illustrate two points rather well. (i) The importance of selecting an appropriate phylogenetic scope for a comparative genomic study (Section 3) – an analysis utilizing legless newts and snakes rather than pufferfish and chickens would not have revealed the ZRS in the first place. (ii) It demonstrates the apparently modular nature of conserved noncoding sequence blocks in evolution – the ZRS can be lost apparently without disrupting the many other functions (OMIM:600725) of *Shh* during vertebrate development.

7 Challenges and Future Directions

There has been great progress in understanding the biology and functions encoded by the human genome since the first draft of a reference sequence was produced in 2001 [59], and much of this insight has been derived through comparison both within and between genomes. However, as with many scientific endeavors, more questions arise with each increment in understanding. For example, we have now realized that much of the functionally constrained sequence in the human genome does not encode proteins and our current understanding of these elements is poor. They are the "dark matter" of the genome. A major and current challenge is to identify each of these elements and to start dissecting their function. In particular, it is likely that they will harbor polymorphisms that impact human health, contributing to common disease susceptibility. The integration of comparative genomics with genetic variation data [1] to identify functional polymorphisms is likely to be a rapidly expanding field with the combined assets of multiple mammalian genome sequences and high-density confirmed polymorphism data available.

Sequence comparison alone may be able to identify all constrained sites, but it is unlikely to be able to establish their associated functions. Rather, it is the synergy of comparative studies with laboratory experiment that provides greatest insight. This approach is embodied by the Encyclopedia of DNA elements (ENCODE) project, an international initiative with the aim of identifying all functional elements in the human genome [35], in effect to shed light on the dark matter of the genome. This is an ambitious and relatively long-term goal. As a first step, a pilot project has been undertaken to investigate 30 Mb of the human genome (approximately 1%, randomly selected) in great detail, applying a broad spectrum of experimental and computational methods to identify functionally important sites (http://genome.gov/10005107). These rigorously annotated regions will be important training and testing grounds for the development of methods in comparative genomics. The UCSC genome browser (http://genome.ucsc.edu/ENCODE/) provides a key portal to access the ENCODE pilot project data.

8 Conclusion

In the middle of 2000, credible estimates of the total number of human protein-coding genes plummeted from 80 000–100 000 to 30 000 or so [90]. These lower counts were essentially confirmed by the early analyses of the human genome [59] and, if anything, the real numbers are likely to be smaller still [50]. Although it is difficult, and perhaps even of little value, to interpret these results within the commonly perceived frameworks of organismal complexity, the fact remains that they have created a new impetus for looking beyond protein-coding genes towards other classes of functional elements, such as noncoding RNAs and, in particular, the *cis*-acting elements regulating gene expression. At the same time, it is sobering to reflect on how unanticipated these downward revisions of gene count were and, accordingly, to reserve judgment on exactly how many more functional elements of major relevance we may expect to find. The methods and early results presented in this review are merely the first steps on a long path towards a broader understanding of the totality of information encoded in the genome.

References

1 INTERNATIONAL HAPMAP CONSORTIUM. 2004. Integrating ethics and science in the International HapMap Project. Nat. Rev Genet. **5**: 467–75.

2 ALTSCHUL, S. F., T. L. MADDEN, A. A. SCHAFFER, J. ZHANG, Z. ZHANG, W. MILLER AND D. J. LIPMAN. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**: 3389–402.

3 AMBROS, V. 2004. The functions of animal microRNAs. Nature **431**: 350–5.

4 APARICIO, S., J. CHAPMAN, E. STUPKA, et al. 2002. Whole-genome shotgun assembly and analysis of the genome of *Fugu rubripes*. Science **297**: 1301–10.

5 BATZOGLOU, S. 2005. The many faces of sequence alignment. Brief. Bioinform. **6**: 6–22.

6 BATZOGLOU, S., L. PACHTER, J. P. MESIROV, B. BERGER AND E. S. LANDER.

2000. Human and mouse gene structure: comparative analysis and application to exon prediction. Genome Res. **10**: 950–8.

7 BEJERANO, G., M. PHEASANT, I. MAKUNIN, S. STEPHEN, W. J. KENT, J. S. MATTICK AND D. HAUSSLER. 2004. Ultraconserved elements in the human genome. Science **304**: 1321–5.

8 BIRD, A. 2002. DNA methylation patterns and epigenetic memory. Genes Dev. **16**: 6–21.

9 BIRD, A., M. TAGGART, M. FROMMER, O. J. MILLER AND D. MACLEOD. 1985. A fraction of the mouse genome that is derived from islands of nonmethylated, CpG-rich DNA. Cell **40**: 91–9.

10 BLANCHETTE, M., W. J. KENT, C. RIEMER, ET AL. 2004. Aligning multiple genomic sequences with the threaded blockset aligner. Genome Res. **14**: 708–15.

11 BOFFELLI, D., J. MCAULIFFE, D. OVCHARENKO, K. D. LEWIS, I. OVCHARENKO, L. PACHTER AND E. M. RUBIN. 2003. Phylogenetic shadowing of primate sequences to find functional regions of the human genome. Science **299**: 1391–4.

12 BRAY, N., I. DUBCHAK AND L. PACHTER. 2003. AVID: a global alignment program. Genome Res. **13**: 97–102.

13 BRAY, N. AND L. PACHTER. 2003. MAVID multiple alignment server. Nucleic Acids Res. **31**: 3525–6.

14 BRENNER, S. E., C. CHOTHIA AND T. J. HUBBARD. 1998. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. Proc. Natl Acad. Sci. USA **95**: 6073–8.

15 BRUDNO, M., M. CHAPMAN, B. GOTTGENS, S. BATZOGLOU AND B. MORGENSTERN. 2003. Fast and sensitive multiple alignment of large genomic sequences. BMC Bioinformatics **4**: 66.

16 BRUDNO, M., C. B. DO, G. M. COOPER, M. F. KIM, E. DAVYDOV, E. D. GREEN, A. SIDOW AND S. BATZOGLOU. 2003. LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. Genome Res. **13**: 721–31.

17 BRUDNO, M., S. MALDE, A. POLIAKOV, C. B. DO, O. COURONNE, I. DUBCHAK AND S. BATZOGLOU. 2003. Glocal alignment: finding rearrangements during alignment. Bioinformatics **19 (Suppl. 1)**: i54–62.

18 BRUDNO, M., R. STEINKAMP AND B. MORGENSTERN. 2004. The CHAOS/DIALIGN WWW server for multiple alignment of genomic sequences. Nucleic Acids Res. **32**: W41–4.

19 CARNINCI, P., T. KASUKAWA, S. KATAYAMA, et al. 2005. The transcriptional landscape of the mammalian genome. Science **309**: 1559–63.

20 CHENG, Z., M. VENTURA, X. SHE, et al. 2005. A genome-wide comparison of recent chimpanzee and human segmental duplications. Nature **437**: 88–93.

21 COLLINS, F. S., E. D. GREEN, A. E. GUTTMACHER AND M. S. GUYER. 2003. A vision for the future of genomics research. Nature **422**: 835–47.

22 CHIMPANZEE SEQUENCING AND ANALYSIS CONSORTIUM. 2005. Initial sequence of the chimpanzee genome and comparison with the human genome. Nature **437**: 69–87.

23 COOPER, G. M., M. BRUDNO, E. D. GREEN, S. BATZOGLOU AND A. SIDOW. 2003. Quantitative estimates of sequence divergence for comparative analyses of mammalian genomes. Genome Res. **13**: 813–20.

24 COOPER, G. M., M. BRUDNO, E. A. STONE, I. DUBCHAK, S. BATZOGLOU AND A. SIDOW. 2004. Characterization of evolutionary rates and constraints in three mammalian genomes. Genome Res. **14**: 539–48.

25 COURONNE, O., A. POLIAKOV, N. BRAY, T. ISHKHANOV, D. RYABOY, E. RUBIN, L. PACHTER AND I. DUBCHAK. 2003. Strategies and tools for whole-genome alignments. Genome Res. **13**: 73–80.

26 DAVIDSON, H., M. S. TAYLOR, A. DOHERTY, A. C. BOYD AND D. J. PORTEOUS. 2000. Genomic sequence analysis of *Fugu rubripes* CFTR and flanking genes in a 60 kb region conserving synteny with 800 kb of human chromosome 7. Genome Res. **10**: 1194–203.

27 Dermitzakis, E. T., E. Kirkness, S. Schwarz, E. Birney, A. Reymond and S. E. Antonarakis. 2004. Comparison of human chromosome 21 conserved nongenic sequences (CNGs) with the mouse and dog genomes shows that their selective constraint is independent of their genic environment. Genome Res. **14**: 852–9.

28 Dermitzakis, E. T., A. Reymond, R. Lyle, et al. 2002. Numerous potentially functional but non-genic conserved sequences on human chromosome 21. Nature **420**: 578–82.

29 Duret, L., F. Dorkeld and C. Gautier. 1993. Strong conservation of non-coding sequences during vertebrates evolution: potential involvement in post-transcriptional regulation of gene expression. Nucleic Acids Res. **21**: 2315–22.

30 Eddy, S. R. 2005. A model of the statistical power of comparative genome sequence analysis. PLoS Biol. **3**: e10.

31 Edgar, A. J. 2005. Mice have a transcribed L-threonine aldolase/GLY1 gene, but the human GLY1 gene is a non-processed pseudogene. BMC Genomics **6**: 32.

32 Eichler, E. E. 2006. Widening the spectrum of human genetic variation. Nat. Genet. **38**: 9–11.

33 Elnitski, L., B. Giardine, P. Shah, et al. 2005. Improvements to GALA and dbERGE II: databases featuring genomic sequence alignment, annotation and experimental results. Nucleic Acids Res. **33**: D466–70.

34 Emison, E. S., A. S. McCallion, C. S. Kashuk, et al. 2005. A common sex-dependent mutation in a *RET* enhancer underlies Hirschsprung disease risk. Nature **434**: 857–63.

35 ENCODE Project Consortium 2004. The ENCODE (ENCyclopedia Of DNA Elements) Project. Science **306**: 636–40.

36 Felsenstein, J. and G. A. Churchill. 1996. A Hidden Markov Model approach to variation among sites in rate of evolution. Mol. Biol. Evol. **13**: 93–104.

37 Gabriel, S. B., R. Salomon, A. Pelet, et al. 2002. Segregation at three loci explains familial and population risk in Hirschsprung disease. Nat. Genet. **31**: 89–93.

38 Gaffney, D. J. and P. D. Keightley. 2005. The scale of mutational variation in the murid genome. Genome Res. **15**: 1086–94.

39 Giardine, B., L. Elnitski, C. Riemer, I. Makalowska, S. Schwartz, W. Miller and R. C. Hardison. 2003. GALA, a database for genomic sequence alignments and annotations. Genome Res. **13**: 732–41.

40 Gibbs, R. A., G. M. Weinstock, M. L. Metzker, et al. 2004. Genome sequence of the Brown Norway rat yields insights into mammalian evolution. Nature **428**: 493–521.

41 Glazko, G. V., E. V. Koonin and I. B. Rogozin. 2005. Molecular dating: ape bones agree with chicken entrails. Trends Genet. **21**: 89–92.

42 Glazov, E. A., M. Pheasant, E. A. McGraw, G. Bejerano and J. S. Mattick. 2005. Ultraconserved elements in insect genomes: a highly conserved intronic sequence implicated in the control of homothorax mRNA splicing. Genome Res. **15**: 800–8.

43 Gogarten, J. P. and L. Olendzenski. 1999. Orthologs, paralogs and genome comparisons. Curr. Opin. Genet. Dev. **9**: 630–6.

44 Hardison, R. C., J. Oeltjen and W. Miller. 1997. Long human–mouse sequence alignments reveal novel regulatory elements: a reason to sequence the mouse genome. Genome Res. **7**: 959–66.

45 Hardison, R. C., K. M. Roskin, S. Yang, et al.2003. Covariation in frequencies of substitution, deletion, transposition, and recombination during eutherian evolution. Genome Res. **13**: 13–26.

46 Hillier, L. W., W. Miller, E. Birney, et al.2004. Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. Nature **432**: 695–716.

47 Hoegg, S., H. Brinkmann, J. S. Taylor and A. Meyer. 2004. Phylogenetic timing of the fish-specific genome duplication correlates with the diversification of teleost fish. J. Mol. Evol. **59**: 190–203.

48 HURST, L. D. 2002. The K_a/K_s ratio: diagnosing the form of sequence evolution. Trends Genet. **18**: 486.

49 IANAKIEV, P., M. J. VAN BAREN, M. J. DALY, et al. 2001. Acheiropodia is caused by a genomic deletion in *C7orf2*, the human orthologue of the *Lmbr1* gene. Am. J. Hum. Genet. **68**: 38–45.

50 IHGSC. 2004. Finishing the euchromatic sequence of the human genome. Nature **431**: 931–45.

51 JAILLON, O., J. M. AURY, F. BRUNET, et al. 2004. Genome duplication in the teleost fish *Tetraodon nigroviridis* reveals the early vertebrate proto-karyotype. Nature **431**: 946–57.

52 KAPITONOV, V. V. AND J. JURKA. 2005. RAG1 core and V(D)J recombination signal sequences were derived from Transib transposons. PLoS Biol. **3**: e181.

53 KAZAZIAN, H. H., JR. 2004. Mobile elements: drivers of genome evolution. Science **303**: 1626–32.

54 KEIGHTLEY, P. D. AND T. JOHNSON. 2004. MCALIGN: stochastic alignment of noncoding DNA sequences based on an evolutionary model of sequence evolution. Genome Res. **14**: 442–50.

55 KEIGHTLEY, P. D., M. J. LERCHER AND A. EYRE-WALKER. 2005. Evidence for widespread degradation of gene control regions in hominid genomes. PLoS Biol. **3**: e42.

56 KENT, W. J. 2002. BLAT – the BLAST-like alignment tool. Genome Res. **12**: 656–64.

57 KENT, W. J. AND A. M. ZAHLER. 2000. Conservation, regulation, synteny, and introns in a large-scale *C. briggsae*–*C. elegans* genomic alignment. Genome Res. **10**: 1115–25.

58 KOWALSKI, P. E., J. D. FREEMAN AND D. L. MAGER. 1999. Intergenic splicing between a HERV-H endogenous retrovirus and two adjacent human genes. Genomics **57**: 371–9.

59 LANDER, E. S., L. M. LINTON, B. BIRREN, et al. 2001. Initial sequencing and analysis of the human genome. Nature **409**: 860–921.

60 LAW, S. Y., M. FOK, S. W. CHENG AND J. WONG. 1992. A comparison of outcome after resection for squamous cell carcinomas and adenocarcinomas of the esophagus and cardia. Surg. Gynecol. Obstet. **175**: 107–12.

61 LETTICE, L. A., S. J. HEANEY, L. A. PURDIE, et al. 2003. A long-range *Shh* enhancer regulates expression in the developing limb and fin and is associated with preaxial polydactyly. Hum. Mol. Genet. **12**: 1725–35.

62 LETTICE, L. A. AND R. E. HILL. 2005. Preaxial polydactyly: a model for defective long-range regulation in congenital abnormalities. Curr. Opin. Genet. Dev. **15**: 294–300.

63 LINDBLAD-TOH, K., C. M. WADE, T. S. MIKKELSEN, et al. 2005. Genome sequence, comparative analysis and haplotype structure of the domestic dog. Nature **438**: 803–19.

64 LUNTER, G., C. P. PONTING AND J. HEIN. 2006. Genome-wide identification of human functional DNA using a neutral indel model. PLoS Comput. Biol. **2**: e5.

65 LYNCH, M. 2006. The origins of eukaryotic gene structure. Mol. Biol. Evol. **23**: 450–68.

66 MA, B., J. TROMP AND M. LI. 2002. PatternHunter: faster and more sensitive homology search. Bioinformatics **18**: 440–5.

67 MANIATIS, T. AND B. TASIC. 2002. Alternative pre-mRNA splicing and proteome expansion in metazoans. Nature **418**: 236–43.

68 MARCHINI, J., P. DONNELLY AND L. R. CARDON. 2005. Genome-wide strategies for detecting multiple loci that influence complex diseases. Nat. Genet. **37**: 413–7.

69 MARGULIES, E. H., M. BLANCHETTE, D. HAUSSLER AND E. D. GREEN. 2003. Identification and characterization of multi-species conserved sequences. Genome Res. **13**: 2507–18.

70 MARGULIES, E. H., J. P. VINSON, W. MILLER, et al. 2005. An initial strategy for the systematic identification of functional elements in the human genome by low-redundancy comparative sequencing. Proc. Natl Acad. Sci. USA **102**: 4795–800.

71 MARTIN, J., C. HAN, L. A. GORDON, et al. 2004. The sequence and analysis of duplication-rich human chromosome 16. Nature **432**: 988–94.

72 MATTICK, J. S. 2004. RNA regulation: a new genetics? Nat. Rev Genet. **5**: 316–23.

73 MAYOR, C., M. BRUDNO, J. R. SCHWARTZ, A. POLIAKOV, E. M. RUBIN, K. A. FRAZER, L. S. PACHTER AND I. DUBCHAK. 2000. VISTA: visualizing global DNA sequence alignments of arbitrary length. Bioinformatics **16**: 1046–7.

74 MILES, C., G. ELGAR, E. COLES, D. J. KLEINJAN, V. VAN HEYNINGEN AND N. HASTIE. 1998. Complete sequencing of the *Fugu* WAGR region from WT1 to PAX6: dramatic compaction and conservation of synteny with human chromosome 11p13. Proc. Natl Acad. Sci. USA **95**: 13068–72.

75 MILLER, W., K. D. MAKOVA, A. NEKRUTENKO AND R. C. HARDISON. 2004. Comparative genomics. Annu. Rev. Genomics Hum. Genet. **5**: 15–56.

76 MORGENSTERN, B. 2004. DIALIGN: multiple DNA and protein sequence alignment at BiBiServ. Nucleic Acids Res. **32**: W33–6.

77 MULLIKIN, J. C. AND Z. NING. 2003. The phusion assembler. Genome Res. **13**: 81–90.

78 NEEDLEMAN, S. B. AND C. D. WUNSCH. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. **48**: 443–53.

79 NEI, M. AND S. KUMAR. 2000. *Molecular Evolution and Phylogenetics*. Oxford University Press, New York, NY.

80 NGUYEN, D. Q., C. WEBBER AND C. P. PONTING. 2006. Bias of selection on human copy-number variants. PLoS Genet. **2**: e20.

81 NIELSEN, R., C. BUSTAMANTE, A. G. CLARK, et al. 2005. A scan for positively selected genes in the genomes of humans and chimpanzees. PLoS Biol. **3**: e170.

82 NING, Z., A. J. COX AND J. C. MULLIKIN. 2001. SSAHA: a fast search method for large DNA databases. Genome Res. **11**: 1725–9.

83 OHTA, T. 1976. Simple model for treating evolution of multigene families. Nature **263**: 74–6.

84 OPHIR, R. AND D. GRAUR. 1997. Patterns and rates of indel evolution in processed pseudogenes from humans and murids. Gene **205**: 191–202.

85 PETROV, D. A. AND D. L. HARTL. 1998. High rate of DNA loss in the *Drosophila melanogaster* and *Drosophila virilis* species groups. Mol. Biol. Evol. **15**: 293–302.

86 PIGANEAU, G. AND A. EYRE-WALKER. 2003. Estimating the distribution of fitness effects from DNA sequence data: implications for the molecular clock. Proc. Natl Acad. Sci. USA **100**: 10335–40.

87 PODLAHA, O. AND J. ZHANG. 2004. Nonneutral evolution of the transcribed pseudogene Makorin1-p1 in mice. Mol. Biol. Evol. **21**: 2202–9.

88 POLLARD, D. A., C. M. BERGMAN, J. STOYE, S. E. CELNIKER AND M. B. EISEN. 2004. Benchmarking tools for the alignment of functional noncoding DNA. BMC Bioinformatics **5**: 6.

89 POTTER, S. C., L. CLARKE, V. CURWEN, et al. 2004. The Ensembl analysis pipeline. Genome Res. **14**: 934–41.

90 ROEST CROLLIUS, H., O. JAILLON, A. BERNOT, et al. 2000. Estimate of human gene number provided by genome-wide analysis using *Tetraodon nigroviridis* DNA sequence. Nat. Genet. **25**: 235–8.

91 ROY, A. M., M. L. CARROLL, S. V. NGUYEN, A. H. SALEM, M. OLDRIDGE, A. O. WILKIE, M. A. BATZER AND P. L. DEININGER. 2000. Potential gene conversion and source genes for recently integrated Alu elements. Genome Res. **10**: 1485–95.

92 SAGAI, T., M. HOSOYA, Y. MIZUSHINA, M. TAMURA AND T. SHIROISHI. 2005. Elimination of a long-range *cis*-regulatory module causes complete loss of limb-specific *Shh* expression and truncation of the mouse limb. Development **132**: 797–803.

93 SAGAI, T., H. MASUYA, M. TAMURA, et al. 2004. Phylogenetic conservation of a

limb-specific, *cis*-acting regulator of *Sonic hedgehog* (*Shh*). Mamm. Genome **15**: 23–34.

94 SCHWARTZ, S., W. J. KENT, A. SMIT, et al. 2003. Human–mouse alignments with BLASTZ. Genome Res. **13**: 103–7.

95 SCHWARTZ, S., Z. ZHANG, K. A. FRAZER, et al. 2000. PipMaker – a web server for aligning two genomic DNA sequences. Genome Res. **10**: 577–86.

96 SHARP, A. J., D. P. LOCKE, S. D. MCGRATH, et al. 2005. Segmental duplications and copy-number variation in the human genome. Am. J. Hum. Genet. **77**: 78–88.

97 SHARPE, J., L. LETTICE, J. HECKSHER-SORENSEN, M. FOX, R. HILL AND R. KRUMLAUF. 1999. Identification of *sonic hedgehog* as a candidate gene responsible for the polydactylous mouse mutant Sasquatch. Curr. Biol. **9**: 97–100.

98 SHE, X., Z. JIANG, R. A. CLARK, et al. 2004. Shotgun sequence assembly and recent segmental duplications within the human genome. Nature **431**: 927–30.

99 SHEMESH, R., A. NOVIK, S. EDELHEIT AND R. SOREK. 2006. Genomic fossils as a snapshot of the human transcriptome. Proc. Natl Acad. Sci. USA **103**: 1364–9.

100 SIEPEL, A., G. BEJERANO, J. S. PEDERSEN, et al. 2005. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. Genome Res. **15**: 1034–50.

101 SINGH, N. D., P. F. ARNDT AND D. A. PETROV. 2005. Genomic heterogeneity of background substitutional patterns in *Drosophila melanogaster*. Genetics **169**: 709–22.

102 SONNHAMMER, E. L. AND R. DURBIN. 1995. A dot-matrix program with dynamic threshold control suited for genomic DNA and protein sequence analysis. Gene **167**: GC1–10.

103 STOYE, J., D. EVERS AND F. MEYER. 1998. Rose: generating sequence families. Bioinformatics **14**: 157–63.

104 SVED, J. AND A. BIRD. 1990. The expected equilibrium of the CpG dinucleotide in vertebrate genomes under a mutation model. Proc. Natl Acad. Sci. USA **87**: 4692–6.

105 TAGLE, D. A., B. F. KOOP, M. GOODMAN, J. L. SLIGHTOM, D. L. HESS AND R. T. JONES. 1988. Embryonic epsilon and gamma globin genes of a prosimian primate (*Galago crassicaudatus*). Nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. J. Mol. Biol. **203**: 439–55.

106 TATUSOV, R. L., N. D. FEDOROVA, J. D. JACKSON, et al. 2003. The COG database: an updated version includes eukaryotes. BMC Bioinformatics **4**: 41.

107 TAYLOR, M. S., R. S. DEVON, J. K. MILLAR AND D. J. PORTEOUS. 2003. Evolutionary constraints on the Disrupted in Schizophrenia locus. Genomics **81**: 67–77.

108 TAYLOR, M. S., C. P. PONTING AND R. R. COPLEY. 2004. Occurrence and consequences of coding sequence insertions and deletions in mammalian genomes. Genome Res. **14**: 555–66.

109 TEAGUE, P. W., M. A. ALDRED, M. JAY, et al. 1994. Heterogeneity analysis in 40 X-linked retinitis pigmentosa families. Am. J. Hum. Genet. **55**: 105–11.

110 THOMAS, J. W., J. W. TOUCHMAN, R. W. BLAKESLEY, et al. 2003. Comparative analyses of multi-species sequences from targeted genomic regions. Nature **424**: 788–93.

111 VAN HEYNINGEN, V. AND K. A. WILLIAMSON. 2002. *PAX6* in sensory development. Hum. Mol. Genet. **11**: 1161–7.

112 VENTER, J. C., M. D. ADAMS, E. W. MYERS, et al. 2001. The sequence of the human genome. Science **291**: 1304–51.

113 VERVOORT, R., A. LENNON, A. C. BIRD, et al. 2000. Mutational hot spot within a new RPGR exon in X-linked retinitis pigmentosa. Nat. Genet. **25**: 462–6.

114 WATERSTON, R. H., K. LINDBLAD-TOH, E. BIRNEY, et al. 2002. Initial sequencing and comparative analysis of the mouse genome. Nature **420**: 520–62.

115 WEBER, J. L. AND E. W. MYERS. 1997. Human whole-genome shotgun sequencing. Genome Res. **7**: 401–9.

116 WENDL, M. C. AND S. P. YANG. 2004. Gap statistics for whole genome shotgun

DNA sequencing projects. Bioinformatics **20**: 1527–34.

117 WOLFE, K. H., P. M. SHARP AND W. H. LI. 1989. Mutation rates differ among regions of the mammalian genome. Nature **337**: 283–5.

118 WOOLFE, A., M. GOODSON, D. K. GOODE, et al. 2005. Highly conserved non-coding sequences are associated with vertebrate development. PLoS Biol. **3**: e7.

119 ZHANG, Z., N. CARRIERO AND M. GERSTEIN. 2004. Comparative analysis of processed pseudogenes in the mouse and human genomes. Trends Genet. **20**: 62–7.

38
Association Studies of Complex Diseases

Momiao Xiong and Li Jin

1 Introduction

Most complex diseases, including obesity, diabetes, cardiovascular disease, hypertension, asthma, psychiatric illness, cancer and inflammatory disease, are common diseases and hence pose great public health concerns [19]. Health states of individuals are a complex, multidimensional phenomenon. Clinical manifestations arise from integrated actions of multiple genetic and environmental factors, through dynamic, epigenetic and regulatory mechanisms [90, 91]. For example, outcome, severity and progression of ankylosing spondylitis (AS) are influenced by genetic factors, environmental (i.e. nongenetic) factors, psychosocial constructs, treatments, biologic constructs (intermediate phenotypes) such as disease activity, possibly biomarkers themselves, and their interactions. It is assumed that AS is the effect of inflammation leading to ankylosis. The susceptibility genes, environments and their interactions will initiate inflammatory arthritis and enthesitis. The subsequent immune response over time slowly leads to varying degrees of bony ankylosis and, later, to spinal fusion. In some patients, the occurrence of spinal arthritis may be accompanied by peripheral arthritis and other nonspinal manifestations. Subsequently, the severity and persistence of axial and/or joint inflammation will affect the rate and extent of ankylosis. The occurrence of ankylosis, in turn, will lead to the long-term health outcomes – spinal fusion, functional disability and work disability. In general, inflammation results in functional disability independent of ankylosis, while ankylosis is the primary cause of spinal fusion.

Therefore, clinical phenotype can be thought of as a synthesis of genes, gene–gene interactions and gene–environment interactions [14, 40, 78, 81]. A general disease model can be represented by Figure 1.

The general disease model assumes that multiple modules of phenotypes, a set of genes and a set of environments, contribute to the outcome of the disease. A module of phenotypes consists of a number of phenotypes which are influenced by the genes and environments. The genes and environments

Bioinformatics - From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

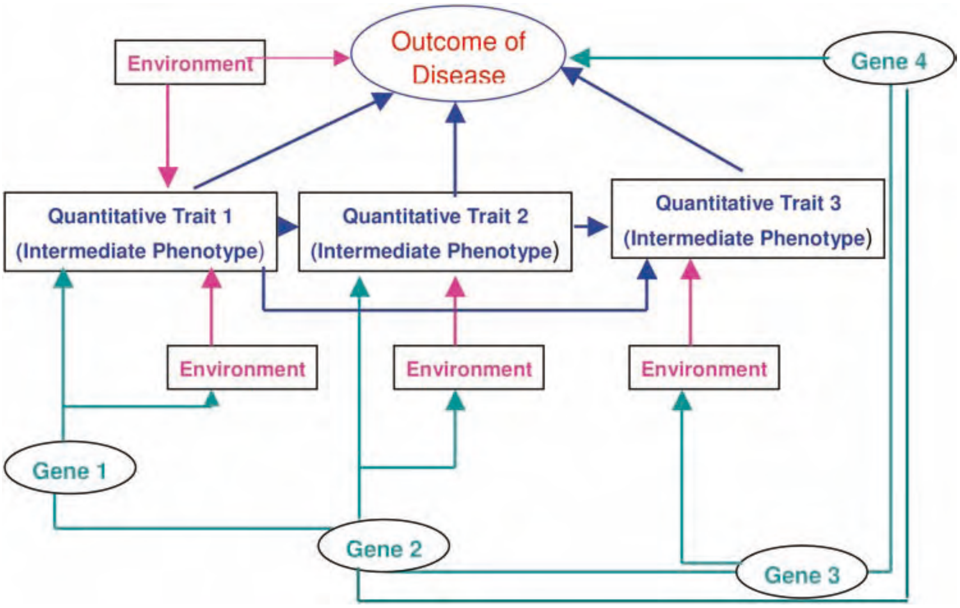

Outcome of Disease
Environment
Gene 4
Quantitative Trait 1 (Intermediate Phenotype)
Quantitative Trait 2 (Intermediate Phenotype)
Quantitative Trait 3 (Intermediate Phenotype)
Environment
Environment
Environment
Gene 1
Gene 2
Gene 3

Figure 1 Scheme of a general disease model.

can be classified into four categories: (i) the genes and environments directly influencing a phenotype, (ii) the genes and environments influencing several phenotypes in a module of phenotypes, (iii) the genes and environments simultaneously influencing several modules of phenotypes, and (iv) the genes and environments directly influencing the outcome of the disease. Therefore, the genes and environments will have direct and indirect effects on the disease. The genes and environments which affect the disease through influencing the phenotypes in the modules will only have indirect effects on the disease. Therefore, the proposed disease model is based hierarchically organized networks of phenotypes, genes and environments.

In the proposed disease model, intermediate phenotypes play an important role. In the context of complex traits, it is often insufficient to consider individuals who merely show a particular clinical symptom as being “affected” [100]. The clinical symptoms might be accompanied by several intermediate phenotypes, each of which is caused by a set of genes, environments and their interactions. Definition of the phenotype is a key issue in dissecting the genetic structure of complex diseases [81]. A narrowly defined disease phenotype beforehand will result in the collection of samples that are genetically more homogeneous and, thus, can offer advantages over broad definitions. Smaller samples with a precise phenotype are more valuable and powerful than a large number of poorly characterized samples [14]. The intermediate phenotypes are simpler in the sense that the number of genetic and environmental factors influencing each intermediate phenotype is presumably smaller than the number of factors affecting the clinical outcome. Therefore, if we can localize the genes for each of the intermediate phenotypes separately, we can characterize most of the genetic susceptibility genes for the complex disease [14, 100].

In recent decades, linkage analyses have been the primary method for genetic studies of diseases. Linkage analysis tests for the cosegregation of a genetic marker and a disease phenotype using family data. A significant linkage result implies that a marker and a susceptibility gene are genetically linked. Linkage analysis has been highly successful for many rare single-gene disorders [53].

However, the fact that many diseases are caused by multiple mutations and genes that individually contribute only modestly to disease risk limits the power of linkage studies. Furthermore, linkage analysis requires multiplex families with multiple affected relatives, which are not available on many occasions. An alternative method to linkage analysis for genetic studies of diseases is association studies that examine the co-occurrence of a marker and disease at the population level [74, 80, 82, 92]. Association analysis has higher power than linkage studies to detect small effects. Common complex diseases involve many genes, most of which have small effects. This fact, together with the imminent completion of the HapMap Project providing a comprehensive catalogue of millions of single nucleotide polymorphisms (SNPs) and haplotypes across diverse populations [3, 105], and rapid development of high-throughput genotyping technologies [9], has increased the importance of association studies in genetic epidemiology [21]. The existence of disease–marker association implies that either the marker allele itself is a functional variant contributing to disease risk or the marker is close enough to the disease locus.

Genetic association studies offer a potentially powerful approach for mapping causal genes with modest effects, but they also raise great challenges in three at least aspects. (i) Since a genome-wide association study involves a large number of SNPs and statistical tests, it is practically impossible to ensure a genome-wide significance level of 0.05 using traditional statistic methods. (ii) Most phenotypic variations are generated by integrated actions of multiple genetic and environmental factors through complex (primarily nonadditive and nonlinear) interactions between genes, and between gene and environments. Detecting interactions among genes or SNP markers is a daunting task. (iii) Most existing analytic methods analyze each genetic marker (or haplotype) and phenotype individually, and do not consider network structures among multiple phenotypes and multiple markers. In short, statistical and computational methods for genetic studies have not kept pace with data collection in the laboratory. Therefore, new techniques need to be

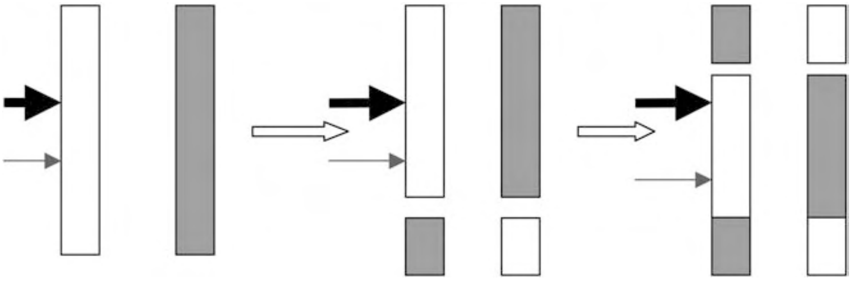


Figure 2 Illustration of generation of LD.

proposed to address these challenging tasks. In this chapter we will focus on how to develop statistical methods which have enough power to ensure a genome-wide significance level.

2 Linkage Disequilibrium (LD), Haplotype and Association Studies

DNA variations are responsible for phenotypic variations. Direct association studies or indirect association studies (nonrandom association with neighboring markers, called LD) can be used to reveal the relationship between DNA variations and phenotypic variations.

2.1 Concepts of LD

LD refers to the nonrandom association of alleles at different marker loci. It is also called *allelic association* or *gametic disequilibrium*. LD is of fundamental importance in genetic studies of complex diseases [50, 82].

LD is due to evolutionary forces in the history of populations such as mutations, recombination and random genetic drift. Suppose that a new mutation arises on individual chromosomes and is flanked by the specific alleles that happen to be present on that chromosome. Over years of transmission of the mutation, through multiple meioses to successive generations, recombination separates the mutation from the original alleles at loci that are unlinked to the mutations. At very closely linked loci, the likelihood of recombination with the disease mutation is low and the original alleles will remain in linkage with the mutation for many generations. By examining the haplotypes at many loci within a large region that does not exhibit recombination, it is sometimes possible to identify a smaller region that appears to be in "LD" with the mutation because the same alleles are present in different families with the diseases. LD provides an indication of which part of the chromosome to study first. See Figure 2.

2.2 Measures of LD

Due to the important implications of LD in association studies of complex diseases, several quantities have been proposed to measure the level of LD between loci, which quantifies the dependence of alleles at two loci. Here, we review several widely used measures of LD.

2.2.1 LD Coefficient D

Consider two marker loci, with two alleles D_1 and d_1 at the first locus and two alleles D_2 and d_2 at the second locus. Assume the population is mating randomly. The disequilibrium coefficient for alleles D_1 and D_2 at two loci is defined as the difference between the haplotype frequency and the product of allele frequencies: $D_{D_1D_2} = P_{D_1D_2} - P_{D_1}P_{D_2}$, where $P_{D_1D_2}$ denotes the frequency of haplotype D_1D_2, P_{D_1} and P_{D_2}, respectively, denote the frequency of allele D_1 at the first locus and the frequency of allele D_2 at the second locus. The maximum likelihood estimate (MLE) of $D_{D_1D_2}$ is estimated by $\hat{D}_{D_1D_2} = \hat{P}_{D_1D_2} - \hat{P}_{D_1}\hat{P}_{D_2}$, where $\hat{P}_{D_1}$ and $\hat{P}_{D_2}$ are the estimated frequency of allele D_1 and D_2, respectively. $P_{D_1D_2}$ is the estimated frequency of haplotype D_1D_2. In practice, the haplotype frequency is unobservable, but can be estimated using one of the algorithms mentioned in the previous section.

2.2.2 Normalized Measure of LD D'

The linkage disequilibrium coefficient $D_{D_1D_2}$ depends on the frequencies of haplotype and alleles, making comparisons between two populations difficult. For the convenience of comparison, Lewontin [61] normalized the above measure of LD by dividing the coefficient D by its maximum value $D_{\max}$, which is given by:

$$\min[P_{A_i}P_{B_j}, (1-P_{A_i})(1-P_{B_j})] \quad \text{if} \quad D_{D_1D_2} < 0$$

$$\min[P_{A_i}(1-P_{B_j}), (1-P_{A_i})P_{B_j}] \quad \text{if} \quad D_{D_1D_2} > 0 \, .$$

This normalized LD measure D' is therefore defined as:

$$D'_{D_1D_2} = \begin{cases} \dfrac{D_{D_1D_2}}{\max(-P_{D_1}P_{D_2}, P_{d_1}P_{d_2})} & D_{D_1D_2} < 0 \\ \dfrac{D_{D_1D_2}}{\min(P_{D_1}P_{d_2}, P_{d_1}P_{D_2})} & D_{D_1D_2} > 0 \end{cases}$$

The normalized LD measure lies between -1 and $+1$, achieving these values when two loci are in complete LD.

2.2.3 Correlation Coefficient *r*

Pearson's correlation coefficient r^2 between two loci is another commonly used measure of the LD. This coefficient is defined as [49]:

$$r^2 = \frac{D^2_{D_1D_2}}{P_{D_1}P_{d_1}P_{D_2}P_{d_2}} \ .$$

r^2 is often used to eliminate the arbitrary sign introduced. When two loci are in linkage equilibrium, r^2 is reduced to zero. There is a simple inverse relationship between this measure and the sample size required to detect association [55].

2.2.4 Composite Measure of LD

LD measures, D, D' and r, assume that individuals mate at random. Under assumption of random mating, the frequency of the genotype is the product of frequencies of haplotypes, and the previously discussed measures of LD can be calculated by estimations of frequencies of alleles and haplotypes which are obtained by MLE. However, this assumption is not always satisfied. When only genotypic data are available and random mating cannot be assumed, the measures of gametic disequilibrium introduced previously cannot be calculated directly. Weir [112] and Weir and Cockerham [113] introduced the following composite measure of LD which combines gametic and nongametic digenic disequilibrium coefficients, and uses only genotype data:

$$\Delta_{AB} = D_{AB} + D_{A/B} = P_{AB} + P_{A/B} - 2P_AP_B \ .$$

This can be calculated as [114]:

$$\Delta_{AB} = \frac{1}{n}\left(2n_{AABB} + n_{AABb} + n_{AaBB} + \frac{1}{2}n_{AaBb}\right) - 2\hat{P}_A\hat{P}_B \ ,$$

where n is the number of individuals sampled, n_{AABB}, n_{AABb}, n_{AaBB} and n_{AaBb} are the numbers of individuals carrying corresponding genotypes at two loci, and $\hat{P}_A$ and $\hat{P}_B$ are the sample frequencies for allele A and B, respectively. The composite measure of LD has the advantage of allowing its determination with genotypic data [46].

2.2.5 The Relationship between the Measure of LD and Physical Distance

Let t denote the age of the mutation which creates the LD. Let θ be the recombination fraction between the two marker loci and $P_{ij}(t)$ be the frequency of the haplotype A_iB_j, at t generations after the mutation causing LD. The haplotype in the next generation is produced either by transmission without recombination or by transmission with recombination between two loci. Thus, we have on average:

$$P_{ij}(t+1) = (1-\theta)P_{ij}(t) + \theta P_{A_i}P_{B_j} \,.$$

Recall that:

$$D_{ij}(t) = P_{ij}(t) - P_{A_i}P_{B_j} \,.$$

Combining above two equations yields the following recursive formula for the calculation of the expectation of the measure of the LD: $D_{ij}(t+1) = (1-\theta)D_{ij}(t)$ and $D_{ij}(t) = D_0(1-\theta)^t$, where $D_0 = P_{ij}(0) - P_{A_i}P_{B_j}$ is an initial measure of LD.

2.3 SNPs and Haplotype Blocks in the Human Genome

2.3.1 SNPs

A SNP is a mismatch between chromosomes in the base present at a particular site in the DNA sequence. It is estimated that about 10 million SNPs (minor allele frequency above 0.1) are present in the human genome, yielding an average density of one in every 200–300 bp [20,58]. These 10 million common SNPs capture 90% of the variation in the population [79] and are widely distributed across the genome, e.g. in exons, introns, intergenic regions, in promoters or enhancers, etc. SNPs play important functional roles. An exonic SNP may directly affect the relevant protein; an intronic SNP can influence the splicing [57]; SNPs in promoters or enhancers can influence gene expression [30]. SNPs are major variations causing diseases.

2.3.2 Tagging SNPs

Millions of SNPs in the human genome provide enough markers to scan the whole genome and their analysis is a powerful tool for genome-wide association studies. However, genotyping a huge number of SNPs is both labor intensive and very expensive. Fortunately, due to the LD between SNPs in a chromosomal region, it is not necessary to type all of the SNPs [115]. Genotyping only a few carefully chosen SNPs which, referred to as "tag SNPs", will provide most of the genetic information in a region with high LD [15,23,42,54] (Figure 3). It was reported from preliminary estimation that most of the genetic variations in the human population could be represented by genotyping 1–2 million tag SNPs across the genome [15, 20, 42]. Thus, using tag SNPs can reduce the number of markers being genotyped without losing much information.

2.3.3 Haplotype Block Model

As a dense set of SNPs markers becomes increasingly available, LD mapping is emerging as a powerful tool for fine mapping of disease susceptibility genes

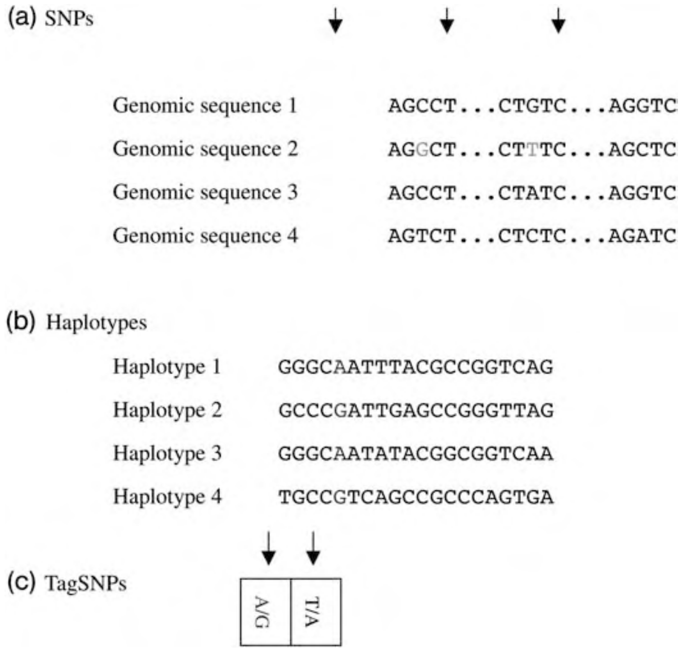

(a) SNPs
Genomic sequence 1 AGCCT...CTGTC...AGGTC
Genomic sequence 2 AGGCT...CTTTC...AGCTC
Genomic sequence 3 AGCCT...CTATC...AGGTC
Genomic sequence 4 AGTCT...CTCTC...AGATC
(b) Haplotypes
Haplotype 1 GGGCAATTTACGCCGGTCAG
Haplotype 2 GCCCGATTGAGCCGGGTTAG
Haplotype 3 GGGCAATATACGGCGGTCAA
Haplotype 4 TGCCGTCAGCCGCCCAGTGA
(c) TagSNPs
A/G
T/A

Figure 3 SNPs, haplotypes and tag SNPs.

and genome-wide association studies. The extent and pattern of LD have been debated for several years. Many evolutionary forces such as mutation, genetic drift, selection, recombination and population bottleneck affect the pattern of LD. It is now widely accepted that the pattern of pairwise LD is erratic. The relationship between the level of pairwise LD and the distance between two individual markers is not monotonic, which complicates LD mapping.

In recent years, there has been growing interest in haplotype and haplotype block LD mapping to alleviate the problem of erratic patterns of pairwise LD. A haplotype is a set of polymorphic alleles that co-occur along the same chromosome. Recent studies showed that haplotypes are not uniformly distributed over the chromosome; rather, they are organized in discrete blocks, in which all pairs of polymorphisms are in strong LD (low recombination), whereas pairs of polymorphisms between blocks show much weaker association (recombination hotspots) [13]. Therefore, a haplotype block shows a largely atomistic pattern and island structure of LD, which greatly simplifies association analyses.

There is strong evidence supporting the existence of haplotype block patterns in the human genome. Daly and coworkers [23] reported block-like patterns in a 500-kb region on chromosome 5 in 129 trios for Crohn's disease. They found that two to four common haplotypes can account for more than 80% of the haplotype variation in their sample. Jeffreys and coworkers [52] studied the haplotype patterns in a region of 216 kb of the major histocompatibility complex (MHC) II complex in the sperm of 50 British males. Their results showed that about 94% of the recombinants in the MHC region were located in the low LD regions, suggesting that the block-like pattern was caused by the recombinant hotspots. Studies by Patil and coworkers [72] revealed similar haplotype block patterns in their samples. Block-like patterns have also been observed on chromosome 22 [24]. More recently, Gabriel and coworkers [38] studied the haplotype patterns across 51 chromosomal regions using population samples from Africa, Europe and Asia. Their findings were consistent with the above observations.

These studies implied that the genome can be portioned into blocks with high LD that are separated by regions with low LD. Within a block, very few common haplotypes or markers can uncover the majority of the DNA variations. These studies also suggested that recombination hotspots and population bottleneck might be possible mechanisms underlying haplotype blocks.

The haplotype block model has important implications for genetic studies. It dramatically alleviates the irregular pattern of LD and holds the promise for mapping complex disease genes. It also provides a simple way of choosing SNPs for large-scale association studies. The main haplotypes in each block could be labeled with a small number of "haplotype-tagging" SNPs (htSNPs) (Figure 3), which would provide an efficient tool for screening each haplotype block region for genetic association studies.

To facilitate genome-wide association studies of complex diseases, the International HapMap Project was initiated in 2003 [20]. Its goal is to determine the common patterns of DNA sequence variation in the human genome, to find SNPs, to construct the general haplotype maps, and to discover the haplotype block structure across the genome and several major populations for all investigators. HapMap is expected to provide a very dense map of SNPs, new tools to dissect genetic structures of complex diseases and to a resource for the development of new treatments. HapMap also holds promise to make the candidate gene-based, linkage-based and genome-wide association studies both practically feasible and cost effective [20].

2.3.4 Definitions of Haplotype Block

Three methods have been proposed to define haplotype blocks. Briefly, the first method defines a block as a region in which "LD decays slowly with distance or not all". Unfortunately, the sampling variation of LD shows considerable fluctuation so that analyses of any trends are made difficult. The second method defines blocks through the optimal partition of a chromosome into a minimum number of blocks and minimum number of representative SNPs. However, there has been no clear biological interpretation of such partitioned haplotype blocks. The third method for defining haplotype block is based on recombination. A haplotype block is defined as a region in which there are no recombination events evidenced in the study sample.

2.3.4.1 Definition of Haplotype Blocks based on Pairwise LD

A simple way to define haplotype blocks is to use pairwise LD [23, 38, 80]. A block is defined whenever all pairwise LD coefficients D' or correlation coefficients r^2 (adjacent and nonadjacent) within a region exceed some predefined threshold, although different thresholds may be used to define the boundary of the blocks in the samples.

Disadvantages of pairwise LD-based blocks include that (i) it is unclear how much the levels of pairwise LD should be chosen to define haplotype blocks which are consistent with the haplotype ancestry, and (ii) the thresholds used to define the block boundaries are subjective and arbitrary. Furthermore, some other factors such as genotyping errors and gene conversion may affect the construction of blocks defined by LD coefficients [107].

2.3.4.2 Definition of Haplotype Blocks based on Haplotype Diversity

The second method for defining haplotype block is based on haplotype diversity and involves the optimal partition of a chromosome into a minimum number of blocks and minimum number of representative SNPs [72, 120]. A block is defined such that the minimum number of tag haplotypes (e.g. two to five) and blocks can account for a maximum proportion of the observations (usually above 80%).

Haplotype-diversity defined blocks share the same concern as those defined by the LD-based method. Thresholds used for determining the number of tag haplotypes which can cover the prespecified proportion of observations are subjective. Also, this method requires known phase data, which are often not accessible or computationally intensive to generate.

2.3.4.3 Definition of Haplotype Blocks based on both Pairwise LD and Haplotype Diversity

Recently, Anderson and Novembre [4] used a minimum-description-length principle to define the best block boundaries. This method simultaneously uses information about pairwise LD as well as the diversity of haplotypes to define haplotype blocks. This method may be better than either of the methods discussed above individually because it uses the joint information on pairwise LD and haplotype diversity, and it is expected to be less severely affected by multiple mutations, gene conversion and genotyping errors. However, this method also requires known phase data, which again mostly depend on statistical inference and are often not accessible in practice.

Haplotype blocks defined by pairwise LD and haplotype diversity are closely related. In general, regions with high pairwise LD usually have low haplotype diversity and regions with high haplotype diversity usually demonstrate low LD. Currently, there is no consensus on which method should be used to define haplotype blocks. Each definition works well for some specific cases. The block structures identified in each study depend strongly on the methods used for defining blocks and on the populations studied. Methods for comparison of haplotype blocks between studies have not been well developed, and it is still unclear which definition of haplotype blocks reflects the underlying biological processes such as recombination and population bottleneck better than others.

2.3.5 **Haplotype Reconstruction**

Although over 10 million SNPs for which the frequency of both alleles exceeds 0.1 exist in the genome [14, 20], analyzing each SNP individually is thought to be less powerful and less informative than simultaneous use of multiple marker information in a region of interest [2, 31, 64, 67]. However, haplotypes are not directly observable. Phase-unknown multilocus genotype data are the primary data sources which we currently have. Although several experimental technologies for molecular haplotyping have been developed [73, 102, 119], these methods are labor intensive, low throughput and costly. Therefore, experimental haplotyping methods are not practically useful for large-scale population studies. Analyzing family data with many relatives is another method to infer haplotypes, but (i) collecting family data is costly and (ii) ambiguity still exists, especially as the number of markers increases. Therefore, computational methods for estimating haplotypes using phase-unknown genotype data offer practical and cost-effective solutions.

2.3.5.1 **Clark's Algorithm**

Clark was the earliest to propose an algorithm based on maximum parsimony to reconstruct haplotypes among unrelated individuals using genotype data [17]. This algorithm first determines the haplotypes from all individuals with no haplotype ambiguity, i.e. the individuals which are complete homozygotes and single-site heterozygotes, assuming Hardy–Weinberg equilibrium (HWE), the basic model of stable frequency distribution among haplotypes in the presence of random mating. Then, the remaining individuals with ambiguous haplotypes are sequentially screened for the possible occurrence of previously recognized haplotypes; the complementary haplotype was then added to the list of resolved haplotypes. Clark's algorithm is straightforward, but does not give unique solutions and does not explicitly assume HWE [69].

2.3.5.2 Expectation Maximization (EM) Algorithm

The EM algorithm infers haplotypes based on maximum likelihood that optimizes the likelihood of occurrence of molecular haplotype frequencies from the observed data, assuming HWE [26,35,76]. The advantages of the EM algorithm include its solid theory, good performance for large samples and relative robustness to departure from HWE. However, since the optimization method is greedy, its performance is sensitive to the initial solution. Inappropriate initial solutions may lead to wrong local maxima, which is serious when there are many distinct haplotypes. Therefore, to ensure finding the global MLE of haplotype frequencies, EM algorithms should be started multiply with several initial solutions. Further, a standard application of the EM algorithm may not be feasible when large number of markers are analyzed simultaneously since the number of haplotypes, and hence the computational time, increase exponentially with the number of markers.

2.3.5.3 Bayesian and Coalescence-based Methods

Stephens and coworkers [95] proposed using a Bayesian approach, either using a simple Dirichlet-prior distribution or a prior distribution which approximates the coalescent, to reconstruct haplotypes from genotype data. The algorithm has been implemented in the program PHASE and its modified versions [62]. This algorithm infers haplotypes based on the following logic – a haplotype that is more similar to the commonly observed haplotype patterns has a higher probability of being present in the population than the less similar haplotypes. The principle for Bayesian haplotype reconstruction methods is to treat the unknown haplotypes as random quantities, and to calculate the *a posteriori* distribution of the unobserved haplotypes given the observed genotype data using prior information and the likelihood. The haplotypes (or haplotype frequencies) can then be estimated from maximizing this *a posteriori* distribution [96]. The prominent feature of this algorithm is the incorporation of coalescence theory into the algorithms and that it outperforms two previously introduce algorithms. The disadvantages include the lack of a measure of overall quality of the inferred haplotypes, slow computation and unclear performance in admixed or rapidly expanding populations when the coalescent model does not hold [69].

Several other coalescence-based and Bayesian algorithms as well as modified EM algorithms have also been developed to facilitate haplotype reconstruction for multiple markers [6, 44, 70].

The algorithms mentioned above have their own advantages and disadvantages. Comparative studies and new methods are still needed for haplotype inference. In addition, methodologies for haplotype reconstruction in large families remain challenging.

2.3.6 Measure of Haplotype Block LD

Suppose that there are k loci within a block. Assume that two alleles A_1 and A_2 at each locus have frequencies P_{A_1} and P_{A_2}, respectively. Consider a k locus haplotype $H_{j_1 j_2 \dots j_k}$ with a sequence of alleles $A_{j_1}, A_{j_2}, \dots, A_{j_k}$, where A_{j_i} at the i-th locus is either A_1 or A_2. Let $P_{H_{j_1 j_2 \cdots j_k}}$ be the population frequency of the haplotype $H_{j_1 j_2 \dots j_k}$. An overall measure of the haplotype LD at the k loci is defined as:

$$\delta_{H_{j_1 j_2 \cdots j_k}} = P_{H_{j_1 j_2 \cdots j_k}} - P_{A_{j_1}} P_{A_{j_2}} \dots P_{A_{j_k}} .$$

This overall measure of the haplotype LD includes the pairwise LD measures and higher-order measures.

Such an overall measure can be applied to measuring LD between a haplotype block and a marker locus. Consider a haplotype $H_{j_1 j_2 \cdots j_k}$ consisting of alleles $A_{j_1} A_{j_2} \dots A_{j_k}$ in the block and an allele M_1 at the marker locus which is outside the block. The haplotype $H_{j_1 j_2 \cdots j_k}$ and the allele M_1form a $(k+1)$ locus haplotype $H_{j_1 j_2 \dots j_k M_1}$. The overall measure of LD for the haplotype $H_{j_1 j_2 \dots j_k M_1}$ can be used to measure LD between the haplotype $H_{j_1 j_2 \cdots j_k}$ and the marker allele M_1, and will be denoted by D_{HM}. Some authors suggested that the haplotype block be treated as alleles and multiallelic analysis for the single marker be applied to the haplotype block analysis [12]. Following this approach, the measure of LD between a haplotype block and the marker locus, denoted by D_{HM}, can be defined as:

$$\begin{aligned} D_{\mathrm{HM}} &= P_{H_{j_1 j_2 \cdots j_k M_1}} - P_{H_{j_1 j_2 \cdots j_k}} P_{M_1} \\ &= P_{H_{j_1 j_2 \cdots j_k M_1}} - (\delta_{H_{j_1 j_2 \cdots j_k}} + P_{A_{j_1}} P_{A_{j_2}} \dots P_{A_{j_k}}) P_{M_1} \\ &= P_{H_{j_1 j_2 \cdots j_k M_1}} - P_{A_{j_1}} P_{A_{j_2}} \dots P_{A_{j_k}} P_{M_1} - P_{M_1} \delta_{H_{j_1 j_2 \cdots j_k}} \\ &= \delta_{\mathrm{HM}} - P_{M_1} \delta_{H_{j_1 j_2 \cdots j_k}} . \end{aligned}$$

This measure of LD between a haplotype and a marker is obtained by removing the haplotype block LD from the LD measure between the haplotype and the marker locus.

3 A General Framework for Population-based Association Studies

3.1 Motivation

Genome-wide association studies are emerging as a promising tool for genetic analysis of complex disease [14, 68, 82, 101]. With the imminent completion of the HapMap Project providing a comprehensive catalogue of common genetic variations in human populations [3], and rapid development of technologies enabling efficient and economical genotyping of a large number of variants [9], genome-wide association studies are becoming practically feasible in the near future [48]. However, one of the major barriers in performing genome-wide association studies is a multiple-testing problem, which may prevent the realization of genome-wide association studies.

Three popular strategies have been used to alleviate multiple-testing problems. One strategy is to focus on controlling false discovery rates, defined as the expected proportion of false positives among the declared significant discovery. A traditional quantity for measuring the overall rate of the multiple tests is the family-wise error rate (FWER), which is the probability of making at least one type I error (a false positive) among all the hypotheses. However, the FWER for ensuring genome-wide significance is too stringent. Recently, another multiple-hypothesis testing error measure, known as the false discovery rate (FDR), has been suggested [7, 22, 33, 83, 97–99]. This is a new notion of global significance for simultaneous testing and is more powerful than FWER. Although FDR is more liberal than FWER, since it allows for controlling the fraction of false discoveries, the p-value of a test still needs to be roughly 5×10^{-6} to ensure FDR ≤ 0.05 if 100 SNPs out of a million SNPs being tested show significance and are left for further investigation [97]. In addition, FDR only deals with how many SNPs should be further investigated, but will not change the order of SNPs ranked by p-values, i.e. FDR cannot detect true linkage or association which the existing statistics may not detect. The second strategy is to construct haplotype blocks by studying LD patterns across the genome and to optimally select a set of robust tag SNPs such that all common variants are either directly genotyped or in strong LD with the genotyped tag SNPs [38, 41, 54, 56, 95, 118, 120, 121]. However, it is unclear whether the haplotype block patterns and tag SNPs are consistent among populations or repeated sampling from within a population. Although the multiple-testing problem can be alleviated by selecting and typing tag SNPs [1, 45], the effect of such a strategy on the significance level is still limited. The third strategy is to adjust p-values. Considering the adjustment for millions of statistical tests, a stringent p-value of 10^{-6} to 10^{-7} has been suggested [37, 68, 104, 109, 111] to ensure a genome-wide significance level of 0.05. Unfortunately, with most existing statistics it is difficult to achieve such stringent p-values. Therefore, developing novel test statistics, which can reach stringent p-values for testing true linkage or association and identify new SNPs showing evidence of linkage or association that are undetected by the traditional statistics, requires immediate consideration.

Before developing a general framework for population-based association studies, we first review the traditional statistics for association studies. The primary assumption for association studies is that a mutation (a disease allele) increases disease susceptibility. Under this assumption, one expects that the disease allele will occur more frequently in the affected individuals (cases) than in the unaffected (controls) [75]. The standard χ^2 test for association studies is to identify the disease locus by comparing the differences in allele/haplotype frequencies between the affected and unaffected individuals. More precisely, the χ^2 statistic is a quadratic form of difference of allele/haplotype frequencies between the affected and unaffected individuals [2,16]. As an alternative to comparing differences in allele/haplotype frequencies, a recently developed class of association tests compares similarities of corresponding genome regions between affected and unaffected individuals [10–12, 25, 106, 119, 121].

A natural way of amplifying differences in frequency is to construct a linear transformation of allele/haplotype frequencies in the currently used statistics for association studies. However, it can be shown that any statistics arising from linear transformation will not change the values of pre-transformation statistics. It is suspected that the relationship between genotype and phenotype is nonlinear for complex diseases [66]. Motivated by this, we propose to use nonlinear transformations of allele/haplotype frequencies in cases (P^A) and in controls (P), i.e. $f(P^A)$and $f(P)$, with the expectation that statistics based on the difference $|f(P^A) - f(P)|$ will only amplify signal, rather than noise and hence they will be more powerful than those based on the difference $|P^A - P|$. For example, the case-control differential may be enhanced with some nonlinear transformations of allele/haplotype frequencies. It can be shown that many similarity -based test statistics are version of the χ^2 test with quadratic transformations of allele or haplotype frequencies. Thus, a general framework for association studies which can unify the allele/haplotype frequency-based association tests and similarity-based association tests can be developed.

3.2 The Traditional χ^2 Test Statistic

Traditional association tests compare the differences in allele or haplotype frequencies between affected and unaffected individuals. Let $n^A = [n_1^A, \ldots, n_m^A]^T$ and $n = [n_1, \ldots, n_m]^T$ be vectors of number of alleles or haplotypes in affected and unaffected individuals, respectively. Let n_A andn_G be the number of sampled affected and unaffected individuals, respectively. Let $P^A = [P_1^A, \ldots, P_m^A]^T$ and $P = [P_1, \ldots, P_m]^T$ be vectors of allele or haplotype frequencies in the affected and unaffected individuals, respectively. Define:

$$\hat{P}_i^A = \frac{n_i^A}{n_A} \quad \text{and} \quad \hat{P}_i = \frac{n_i}{n_G} .$$

Let $\hat{P}^A = [\hat{P}_1^A, \ldots, \hat{P}_m^A]^T$ and $\hat{P} = [\hat{P}_1, \ldots, \hat{P}_m]^T$. By standard statistical theory [60], we know that the vectors n^A and n following multinomial distributions with the following variance-covariance matrix:

$$\prod^{A} = n_A[\mathrm{diag}(P_1^A, \ldots, P_m^A) - P^A(P^A)^{\mathrm{T}}] ,$$

and:

$$\prod = n_G[\mathrm{diag}(P_1, \ldots, P_m) - PP^{\mathrm{T}}] ,$$

respectively. Here, $\mathrm{diag}(P_1^A, \ldots, P_m^A)$ and $\mathrm{diag}(P_1, \ldots, P_m)$ denote diagonal matrices with the diagonal elements $P_1^A, \ldots, P_m^A$ and $P_1, \ldots, P_m$, respectively. Let:

$$\Sigma^A = \mathrm{diag}(P_1^A, \ldots, P_m^A) - P^A(P^A)^{\mathrm{T}} \quad \text{and} \quad \Sigma = \mathrm{diag}(P_1, \ldots, P_m) - PP^{\mathrm{T}} .$$

The allele or haplotype frequencies are asymptotically distributed as the following multivariate normal distribution:

$$N\left(P^A, \frac{1}{n_A}\Sigma^A\right)$$

and:

$$N\left(P, \frac{1}{n_G}\Sigma\right) ,$$

respectively.

One form of the standard χ^2 statistic for case-control association studies is given by:

$$T = (P^A - P)^{\mathrm{T}}\Lambda^-(P^A - P) ,$$

where $\Lambda = \frac{1}{n_A}\Sigma^A + \frac{1}{n_G}\Sigma$ and Λ^- is a generalized inverse of the matrix Λ.

Under the null hypothesis of no association of the marker with the disease, T is asymptotically distributed as a central $\chi^2_{(m-1)}$ distribution.

If we ignore the terms $-P_i^2$ and $-P_iP_j$ $(i, j = 1, \ldots, m)$ in the elements of matrix $_\Sigma$, the variance-covariance matrix Σ is reduced to:

$$\Sigma \approx \begin{bmatrix} P_1 & 0 & \ldots & 0 \\ 0 & P_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & P_m \end{bmatrix} = \mathrm{diag}(P_i) .$$

Similarly, we have $\Sigma^A \approx \mathrm{diag}(P_i^A)$ for the affected individuals. Thus, T can be reduced to:

$$T = \sum_{i=1}^{m} \frac{(P_i - P_i^A)^2}{\frac{P_i}{2n_G} + \frac{P_i^A}{2n_A}} .$$

If we assume that the numbers of affected and unaffected individuals are equal, i.e. $n_A = n_G = n$, then the χ^2 test statistic T can be further reduced to:

$$T = 2n \sum_{i=1}^{m} \frac{(P_i - P_i^A)^2}{P_i + P_i^A} ,$$

which is exactly the formula of the standard χ^2 test statistic [16].

3.3 Test Statistics

From Section 3.2 we know that the principle behind the standard χ^2 test in case-control studies is to compare differences in allele/haplotype frequencies between two populations (e.g. cases and controls). We expect that amplifying such differences may improve the power of detecting disease susceptibility genes. One strategy for amplifying the difference is to nonlinearly transform the frequencies. The difference in the values of nonlinear function of allele/haplotype frequencies between cases and controls should be larger than the difference in original allele/haplotype frequencies. This observation motivates us to develop a general framework for population-based association studies which is based on the difference in nonlinear transformation of allele/haplotype frequencies between cases and controls.

Assume that n_A affected individuals and n_G unaffected individuals are sampled. Let $\hat{P}_{H_i}^A$ and $\hat{P}_{H_i}$ be the estimators of frequencies of haplotype H_i in cases and controls. The allele/haplotype frequencies are asymptotically distributed as the multivariate normal distributions $N(P^A, \frac{1}{2n_A}\Sigma^A)$ and $N(P, \frac{1}{2n_G}\Sigma)$, respectively, where $P^A = [P_{H_1}^A, \ldots, P_{H_m}^A]^{\mathrm{T}}$, $P = [P_{H_1}, \ldots, P_{H_m}]^{\mathrm{T}}$, $\Sigma^A = \mathrm{diag}(P_1^A, \ldots, P_m^A) - P^A(P^A)^{\mathrm{T}}$ and $\Sigma = \mathrm{diag}(P_1, \ldots, P_m) - PP^{\mathrm{T}}$.

Let $f(x)$ be a continuously differentiable nonlinear function with a nonzero differential at x. Let $X_j = f(\hat{P}_{H-j}^A)$ for $j = 1, \ldots, m$, $X = [X_1, \ldots, X_m]^{\mathrm{T}}$, $Y_j = f(\hat{P}_{H_j})$ and $Y = [Y_1, \ldots, Y_m]^{\mathrm{T}}$. Then, the random vectors X and Y are asymptotically distributed as multivariate normal distributions $N(f(P^A), \frac{1}{2N_A}B\Sigma^A B^T)$ and $N(f(P), \frac{1}{2n_G}C\Sigma C^{\mathrm{T}})$, respectively [87], where $b_{ii} = \frac{\partial f(P_{H_i}^A)}{\partial P_{H_i}^A}$, $b_{ij} = 0$, $c_{ii} = \frac{\partial f(P_{H_i})}{\partial P_{H_i}}$, $c_{ij} = 0$, $B = (b_{ij})_{m \times m}$ and $C = (c_{ij})_{m \times m}$.

Define the matrix:

$$\Lambda = \frac{1}{2n_A} B\Sigma^A B^T + \frac{1}{2n_G} C\Sigma C^{\mathrm{T}} .$$

Let $\hat{\Lambda}$ be an estimator of the matrix Λ. We propose the following test statistic T_N to test the association of the alleles/haplotypes with the disease [123]:

$$T_N = (X - Y)^{\mathrm{T}} \hat{\Lambda}^{-} (X - Y) ,$$

where $\hat{\Lambda}^{-}$ is the generalized inverse of the matrix $\hat{\Lambda}$. The null hypothesis is that there is no association of alleles/haplotypes with the disease, i.e. $H_0 : P^A = P$. Let $r = \text{rank}(\hat{\Lambda})$. Under the null hypothesis, T_N is asymptotically distributed as a central χ^2 with r degrees of freedom (Theorem A, p. 122 in Ref. [87]; p. 13 in Ref. [43]). The test statistic T_N defines a class of nonlinear tests. Various nonlinear functions with some regularity can be used to construct the test statistic. Table 1 lists some of the nonlinear functions used in this study and their corresponding derivatives.

Table 1 Some of the nonlinear transformations for allele or haplotype frequencies

Function	Derivative
Entropy	
$-x \log x$	$-1 - \log x$
Exponential	
e^x	e^x
Polynomial	
$x^2 + x + 1$	$2x + 1$
Sigmoid	
$\dfrac{1}{1+\mathrm{e}^{-x}}$	$\dfrac{\mathrm{e}^{-x}}{(1+\mathrm{e}^{-x})^2}$
Gaussian	
$\mathrm{e}^{-\frac{(x-c)^2}{2\sigma^2}}$	$\dfrac{c-x}{\sigma^2} \times \mathrm{e}^{-\frac{(x-c)^2}{2\sigma^2}}$
Reciprocal	
$\dfrac{1}{x}$	$-\dfrac{1}{x^2}$

3.4 Null Distribution of the Nonlinear Statistics

In the previous sections, we have shown that when the sample size is large enough to apply large sample theory, the nonlinear test statistics under null hypothesis of no association are asymptotically distributed as central χ^2 distributions. To examine the validity of this statement, we performed a series of simulation studies [123]. The computer program SNaP [71] was used to generate haplotypes of the sample individuals. Two datasets with a single haplotype block each were simulated. The first dataset has two marker loci which generated four haplotypes with frequencies 0.2952, 0.2562, 0.1957 and 0.2529. The second dataset has six marker loci which generated eight haplotypes with frequencies 0.1820, 0.1461, 0.1406, 0.1291, 0.1211, 0.1107, 0.0817 and 0.0887. For each dataset, 20 000 individuals who were equally divided into cases and controls were generated in the general population.

To examine whether the asymptotic results of the nonlinear test statistics still hold for small sample size under the null hypothesis of no association, 100–500 individuals were randomly sampled from each of the cases and controls. In total, 10 000 simulations were repeated for each of the nonlinear test statistics. In each simulation, the nonlinear test statistics were calculated. Figure 4(A and B) plots the histograms of the nonlinear test statistics based on entropy and exponential functions using two-SNP haplotypes. It can be seen that the null distributions of nonlinear statistics are similar to the theoretical central χ^2 distributions even under the scenario of smaller sample size. Type I error rates of the nonlinear test statistics for sample sizes from 100 to 500 individuals using two-SNP and six-SNP haplotypes have been summarized, and show that the estimated type I error rates (at the significance level 0.05) of the nonlinear test statistics were not appreciably different from the nominal level $\alpha = 0.05$.

3.5 Power of the Nonlinear Test Statistics and the Standard χ^2 Test Statistic

The standard χ^2 test statistic is a special case of the general nonlinear test statistics. The nonlinear test statistics can be used as the general framework for association studies. To further study the merit of the nonlinear statistics for association studies we need to evaluate the performance of nonlinear tests and compare the power of several nonlinear test statistics with that of the standard χ^2 test statistic.

To gain further understanding the power of the nonlinear statistics, we first study their noncentrality parameters. The alternative hypothesis is that there is at least one allele or haplotype associated with the disease, i.e. $H_a : P^A \neq P$. Under the alternative hypothesis, the test statistic T_{N} is asymptotically distributed as a noncentral $\chi^2_{(r)}$ distribution with the following noncentrality parameter:

$$\lambda_{\mathrm{N}} = [f(P^4) - f(P)]^{\mathrm{T}} \Lambda^{-1} [f(P^A) - f(P)] \, ,$$

where:

$$\begin{aligned}
r &= \operatorname{rank}(\Lambda), f(P^A) = [f(P^A_{H_1}), \ldots, f(P^A_{H_m})]^{\mathrm{T}}, \\
f(P) &= [f(P_{H_1}), \ldots, f(P_{H_m})]^{\mathrm{T}} \, , \\
\Lambda &= \frac{1}{2n_A} B \Sigma^A B^{\mathrm{T}} + \frac{1}{2n_G} C \Sigma C^T, \quad \Sigma^A = \operatorname{diag}(P^A_{H_1}, \ldots, P^A_{H_m}) - P^A (P^A)^{\mathrm{T}} \, , \\
\Sigma &= \operatorname{diag}(P_{H_1}, \ldots, P_{H_m}) - P P^{\mathrm{T}} \, , \quad P^A = [P^A_{H_1}, \ldots, P^A_{H_m}]^{\mathrm{T}} \, , \\
P &= [P_{H_1}, \ldots, P_{H_m}]^{\mathrm{T}} \, , \\
b_{ii} &= \frac{\partial f(P^A_{H_i})}{\partial P^A_{H_i}}, b_{ij} = 0, i \neq j, c_{ii} = \frac{\partial f(P_{H_i})}{\partial P_{H_i}}, c_{ij} = 0, i \neq j, B = (b_{ij})_{m \times m} \\
& \text{and } \mathrm{C} = (c_{ij})_{m \times m} \, .
\end{aligned}$$

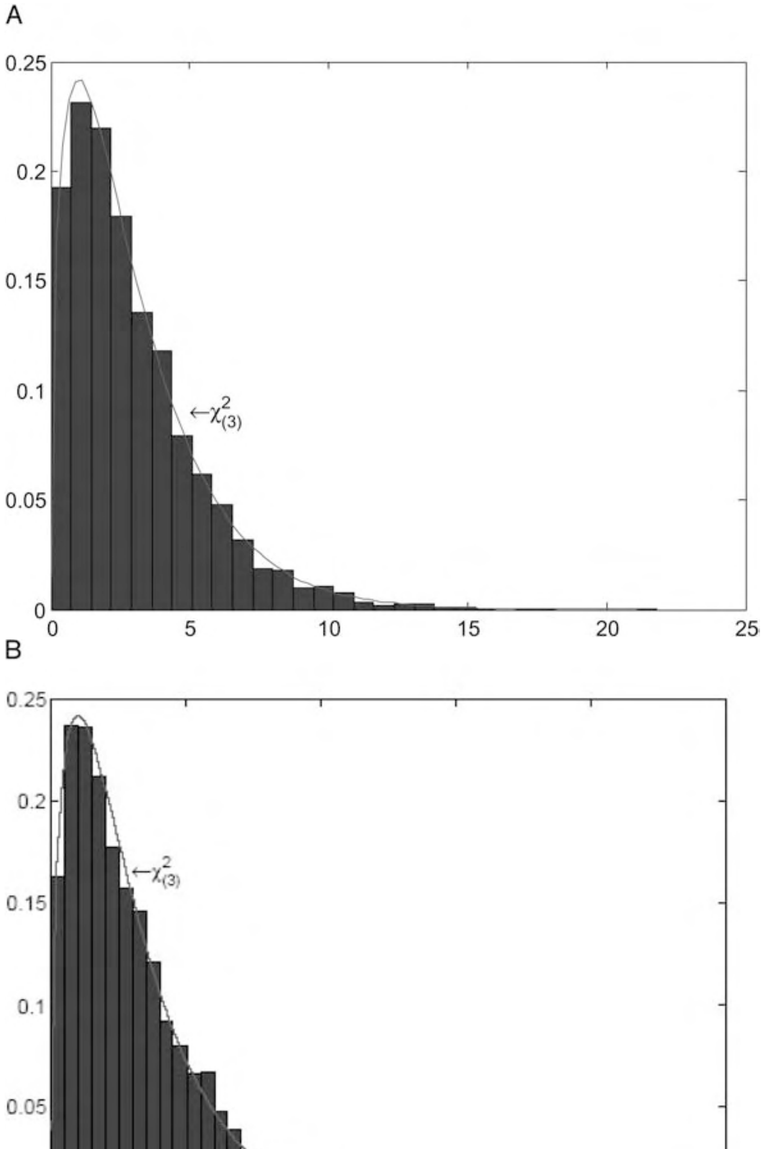


Figure 4 Histograms of nonlinear test statistics.

The noncentrality parameter λ_N can be approximated by (Appendix A) [123]:

$$\lambda_N \approx e^2 \delta_{HD}^T \left(I + \frac{1}{2}S\right)^T \left[\frac{1}{2n_A}(I+S)\Sigma^A(I+S) + \frac{1}{2n_G}\Sigma\right]^- \left(I + \frac{1}{2}S\right)\delta_{HD}\,, \tag{1}$$

where

$$e = \frac{P_D(f_{11} - f_{12}) + P_d(f_{12} - f_{22})}{P(A)}\ , \ \delta_{\mathrm{HD}} = [\delta_{H_1D}, \ldots, \delta_{H_mD}]^{\mathrm{T}}$$

and

$$S = C^{-1}H(P^A - P) = \mathrm{diag}\left(\frac{ef''(P_{H_1})\delta_{H_1D}}{f'(P_{H_1})}, \ldots, \frac{ef''(P_{H_m})\delta_{H_mD}}{f'(P_{H_m})}\right) .$$

The matrix S measures the strength of the nonlinearity of the nonlinear transformation $f(P)$ (Appendix A). Note that under the same alternative hypothesis, the traditional χ^2 test statistic, which is defined as:

$$T = (\hat{P}^A - \hat{P})^T \hat{\Lambda}_0^{-1}(\hat{P}^A - \hat{P})\ , \quad \hat{\Lambda}_0 = \frac{1}{2n_A}\hat{\Sigma}^A + \frac{1}{2n_G}\hat{\Sigma}\ ,$$

is a noncentral $\chi^2_{(r)}$ distribution with the noncentrality parameter:

$$\lambda \approx e^2 \delta_{\mathrm{HD}}^{\mathrm{T}} \left[\frac{1}{2n_A}\Sigma^A + \frac{1}{2n_G}\Sigma\right]^{-} \delta_{\mathrm{HD}}\ . \tag{2}$$

Comparing the noncentrality parameters λ_{N} and λ, we can see that the noncentrality parameter λ_{N} involves one more term S than the noncentrality parameter λ. The matrix S characterizes the nonlinearity of the nonlinear function. The power of the nonlinear test statistics depends on the strength of the nonlinearity of the nonlinear function through the matrix S. The matrix S is referred to as the strength matrix of the nonlinearity of the nonlinear function.

If the product terms of the haplotype frequencies in the variance-covariance matrices Σ^A and Σ are ignored, the matrices Σ^A and Σ can be approximated by $\Sigma^A = \mathrm{diag}(P^A_{H_1}, \ldots, P^A_{H_m})$ and $\Sigma = \mathrm{diag}(P_{H_1}, \ldots, P_{H_m})$. Then the noncentrality parameters λ_{N} andλ will be further reduced to:

$$\begin{aligned} \lambda_{\mathrm{N}} &\approx e^2 \sum_{i=1}^{m} \frac{\delta^2_{H_iD}\left(1 + \frac{e\pi_i}{2}\delta_{H_iD}\right)^2}{\frac{1}{2n_A}\left(1 + \frac{e\pi_i\delta_{H_iD}}{2}\right)^2 P^A_{H_i} + \frac{1}{2n_G}P_{H_i}} \\ \lambda &\approx e^2 \sum_{i=1}^{m} \frac{\delta^2_{H_iD}}{\frac{1}{2n_A}P^A_{H_i} + \frac{1}{2n_G}P_{H_i}}\ , \end{aligned} \tag{3}$$

where $\pi_i = \frac{f''(P_{H_i})}{f'(P_{H_i})}$. The parameter π_i is proportional to the curvature of a nonlinear function [5] and influences the noncentrality parameter λ_{N}.

From the above formulas, we can see that both noncentrality parameters λ and λ_{N} depend on the frequencies of the allele or haplotypes, penetrance, the measure of the LD between the marker alleles or haplotypes and the disease allele, as well as the number of the sampled affected and unaffected individuals. In addition, the noncentrality parameter of the nonlinear test λ_N also depends on the curvature, which measures the degree of nonlinearity of the nonlinear function.

Now we compare the power of several nonlinear test statistics with that of the standard χ^2 test statistic. The nonlinear functions for construction of nonlinear test statistics are listed in Table 1. The markers are assumed to be bi-allelic (i.e. SNPs). Specifically, we consider two scenarios: (i) a disease locus, and (ii) two marker loci and a disease locus which is located in the middle of two markers [123]. The calculation of the power is based on analytic methods which are based on calculation of the noncentrality parameter.

We first investigate the expected noncentrality parameters of the nonlinear tests statistics at the disease locus. We assume that the frequencies of two alleles at the disease locus in controls are both equal to 0.5. Figure 5 plots the expected noncentrality parameters of the nonlinear test statistics and the standard χ^2 test statistic as a function of the frequency of the disease allele in cases. From Figure 5 we can see three remarkable features. (i) The expected noncentrality parameters of all test statistics increase as difference in frequency of the disease allele between cases and controls increases. (ii) The expected noncentrality parameters of the most nonlinear test statistics in Table 1 (except for the statistic based on reciprocal transformation of the allele frequency) are larger than that of the standard χ^2 test statistic. (iii) The expected noncentrality parameters of the nonlinear test statistics in Table 1 (except for the statistic based on reciprocal transformation of the allele frequency) are almost indistinguishable.

We then investigate the power of the nonlinear test statistics at the disease locus. Figure 6(A–C) plots the power of the nonlinear test statistics and the standard χ^2 test statistic as a function of the disease allele frequency under three different disease models: (1) disease model with penetrance $f_{11} = 1$, $f_{12} = 0.2$ and $f_{22} = 0.1$; (2) disease model with penetrance $f_{11} = 1$, $f_{12} = 1$ and $f_{22} = 0.1$, and (3) genotype relative risk model, in which the genotype relative risk for genotypes *Dd* and *DD* is r and r^2 times greater than that of the genotype *dd* [82]. Several features emerge from these figures. (i) The power for most of the nonlinear test statistics is higher than that of the standard χ^2 test statistic, but the power of the test statistic based on the reciprocal function is lower than that of the standard χ^2 test statistic. The power curves of the exponential, sigmoid and quadratic functions are similar. (ii) The power of the nonlinear test statistics is influenced by the disease model. The shapes of the nonlinear test statistics in disease model (2) are different from that of the test statistics in disease models (1) and (3). (iii) The power of the test statistics depends on disease allele frequency. The shapes of the power curves in disease model (1) and (3) are roughly bell-shaped; however, the shapes of the

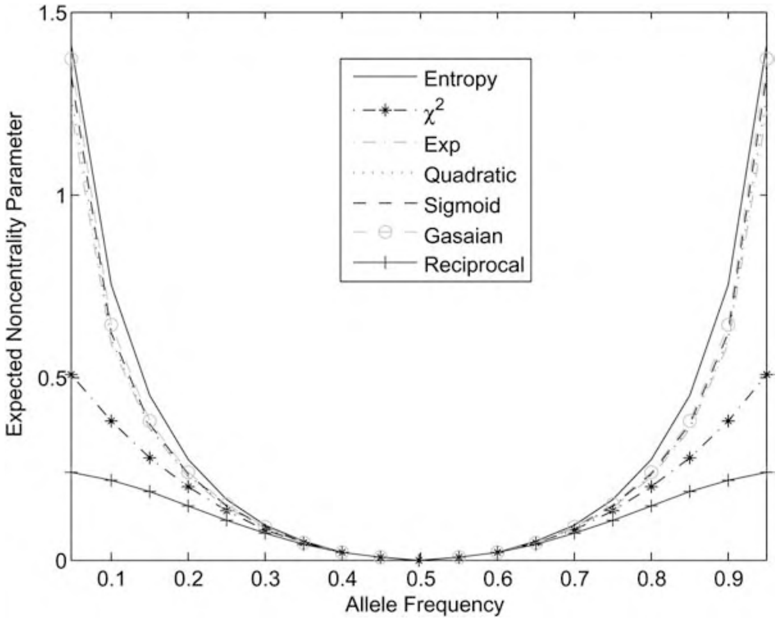


Figure 5 Expected noncentrality parameters as a function of allele frequency.

power curves in disease model (2) are skewed to the left. (iii) The strength of the nonlinearity of the nonlinear transformations also influences the power of the nonlinear test statistics. Figure 7 shows that the measure π_1 of nonlinearity for several nonlinear transformations as a function of disease allele frequency. The measure of the reciprocal function is a negative function of the disease allele frequency and is the smallest among the nonlinear functions being studied. Figure 6(A–C) also shows that the power of the reciprocal-based test statistic is the smallest among the nonlinear tests being studied. In many cases, the larger the measure of nonlinearity of the nonlinear transformation for allele or haplotype frequency, the higher the power of the nonlinear test statistic. However, since the power of the test statistics also depends on other parameters such as disease models, the correlations between the measure of the nonlinearity of the nonlinear transformation and the power of their corresponding nonlinear test statistics follow complex patterns.

Next, we study the power of the nonlinear tests using haplotypes at the marker loci. The markers are assumed to be bi-allelic (i.e. SNP). In particular, we consider two marker loci and a disease locus that is located in the middle of the two markers. The average haplotype frequencies in the affected and unaffected individuals are calculated by equations (1) and (4) in Akey and coworkers [2]. The power of the nonlinear test statistics and the standard χ^2 statistics using four haplotypes generated by two marker loci as a function

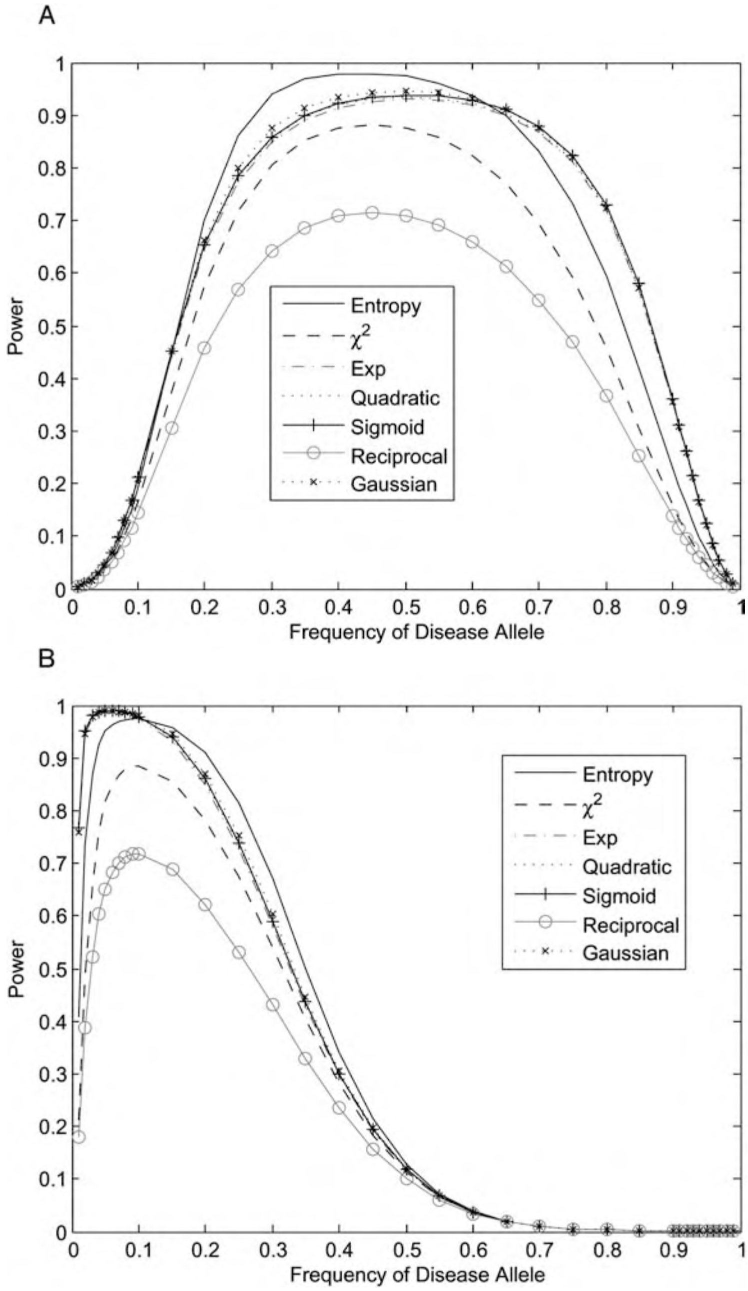


Figure 6 Power of nonlinear test statistics as a function of disease allele frequency.

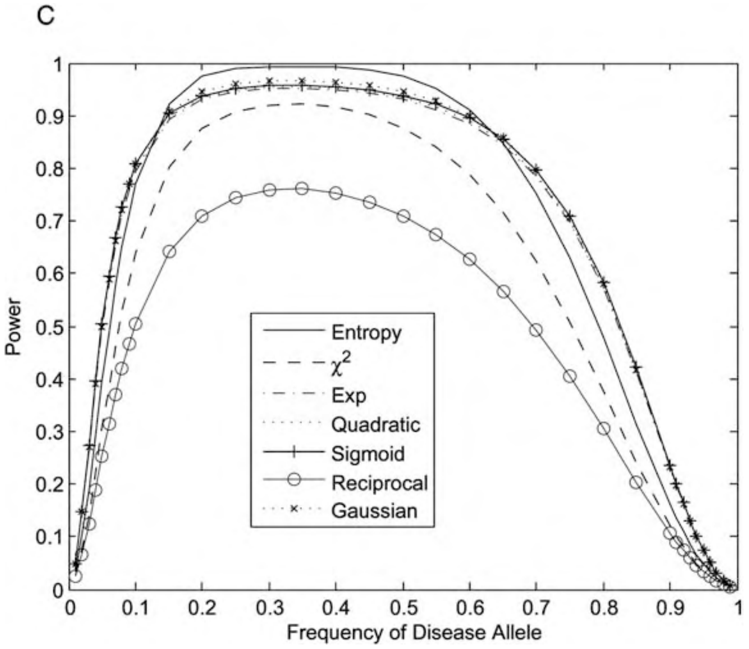


Figure 6 (continued)

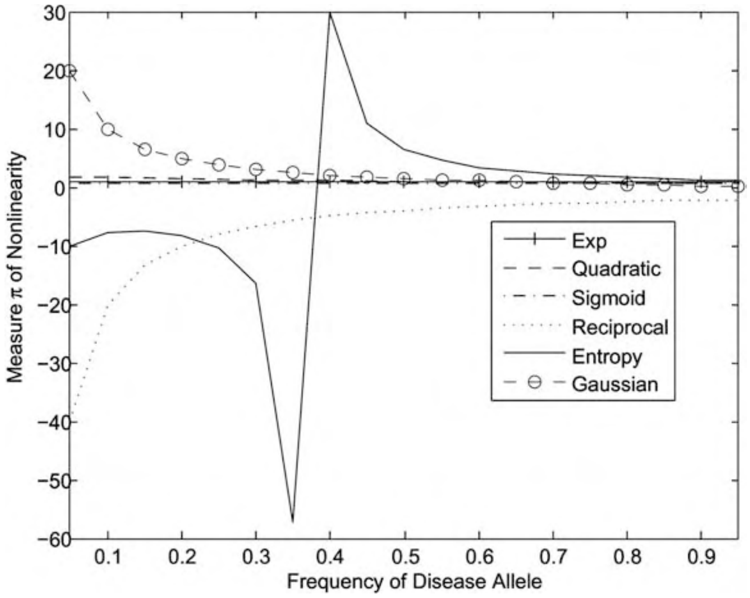


Figure 7 Measure π of nonlinearity as a function of disease allele frequency.

of the genetic distance between the disease locus and its flanking marker loci for recessive, dominance and genotype relative risk disease models is shown in Figure 8(A–C, respectively). The data demonstrate that the power of the nonlinear test statistics is higher than that of the standard χ^2 test, except for the nonlinear test based on the reciprocal. Similar to the scenario at the disease locus, the power of the nonlinear test statistics at the marker loci also depends on the disease model, haplotype frequency, disease allele frequency, the measure of the nonlinearity of the nonlinear functions, and the measure of LD between haplotypes at the marker loci and disease locus.

4 Similarity-based Statistics for Association Studies

We often observe that affected individuals share common haplotypes in the region surrounding disease mutations more often than unaffected individuals [36, 55, 116]. One way to test the excessive sharing of common haplotyes among affected individuals is to compare differences in haplotype frequencies between affected individuals and unaffected individuals [2]. Another way is to compare differences in haplotype similarity between affected and unaffected individuals [10,12,28,106,108]. Before introducing similarity-based statistics for association studies, we will study how to measure the haplotype similarity.

4.1 Similarity Measures

Several haplotype similarity measures have been developed to quantify degrees of haplotype sharing [106]. Here, we introduce three widely used measures of haplotype similarity.

Consider K marker loci which generate m haplotypes. Assume that n_A haplotypes are sampled from the affected individuals and n_G haplotypes are sampled from the unaffected individuals. Let H_i be one of the m haplotypes at the K marker loci. Let P_{H_i} and $P^A_{H_i}$ be the frequency of the haplotype H_i in the unaffected individuals and the affected individuals, respectively. Suppose that the number of H_i haplotypes in the unaffected individuals and affected individuals are n_i and n^A_i, respectively. Let Γ_{H_i} and $\Gamma^A_{H_i}$ be the similarity measure of the haplotype H_i in the unaffected individuals and affected individuals, respectively. A similarity measure of the haplotype H_i is defined as:

$$\Gamma_{H_i} = \frac{n_i}{n_G} \sum_{j=1}^{m} \frac{n_j}{n_G} S(H_i, H_j) = P_{H_i} S_i P \, ,$$

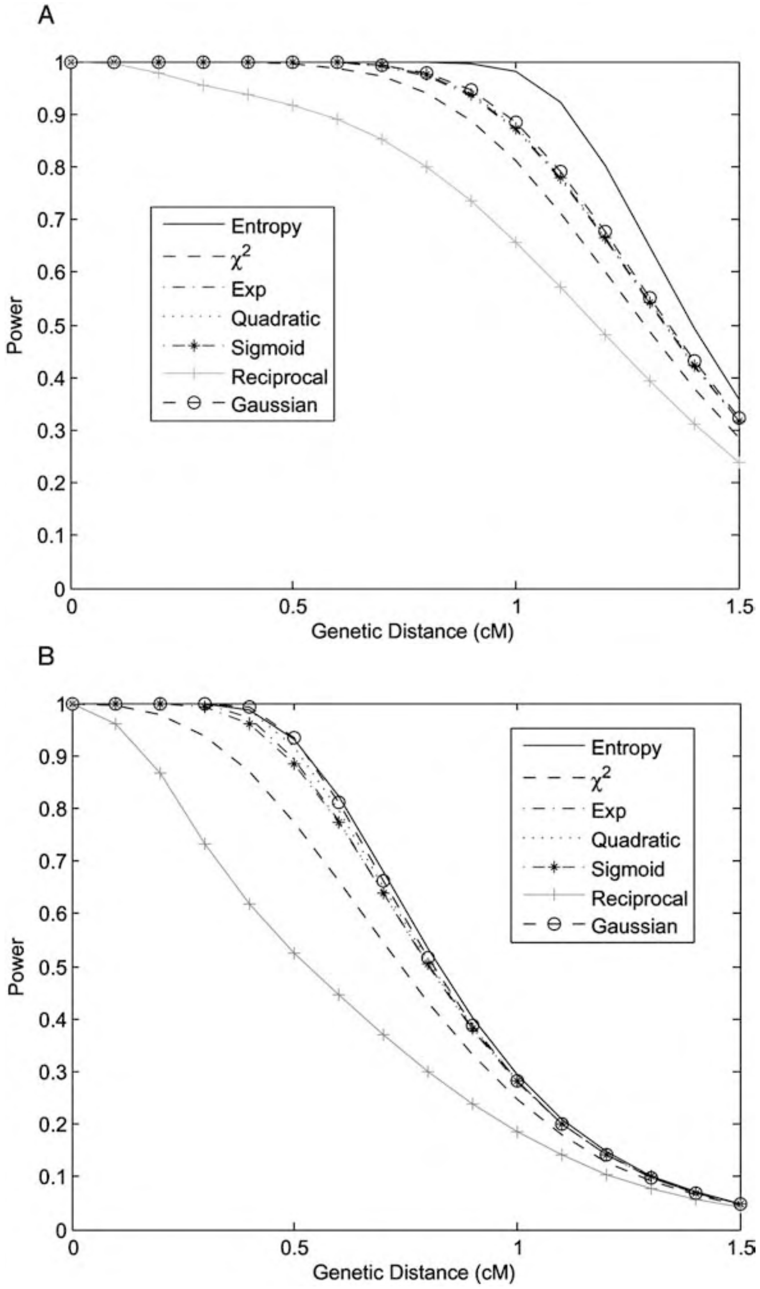


Figure 8 Power of nonlinear test statistics as a function of genetic distance; see text for details.

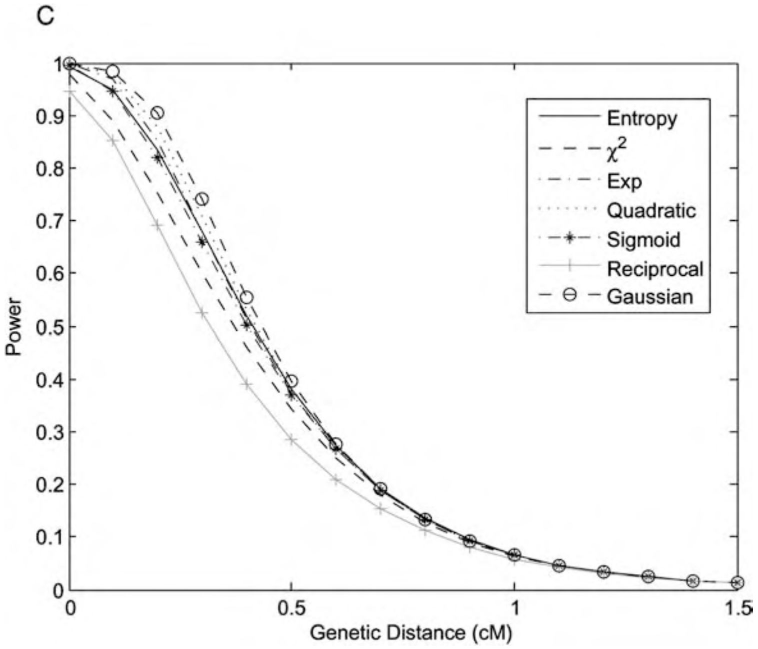


Figure 8 (continued)

where

$$P = [P_{H_1}, \ldots, P_{H_m}]^{\mathrm{T}} \;,\; S_i = [S(H_i, H_1), \ldots, S(H_i, H_m)] \;,\; S = (S(H_i, H_j))_{m \times m} \;,$$

referred to as a similarity matrix, and $S(H_i, H_j)$ is a measure to quantify similarity between the haplotype H_i and the haplotype H_j and will be defined below.

The similarity measure of all the haplotypes in the unaffected individuals, denoted by Γ, is defined as summation of the similarity measure of individual haplotype, i.e., $\Gamma = \sum_{i=1}^{m} \Gamma_{H_i}$. Then, we have $\Gamma = P^{\mathrm{T}}SP$. Similarly, for the affected individuals, we have $\Gamma = (P^A)^T S^A P^A$, where P^A, S^A and Γ^A are similarly defined as that for unaffected individuals.

4.1.1 **Matching Measure**

The matching measure $S(H_i, H_j)$ is defined as:

$$S(H_i, H_j) = \begin{cases} 1 & H_i = H_j \\ 0 & \text{otherwise} \;. \end{cases}$$

Therefore, for the matching measure, S is an identity matrix.

The matching measure considers only identical haplotypes as similar haplotypes and any nonidentical haplotypes are not similar, even if they share a

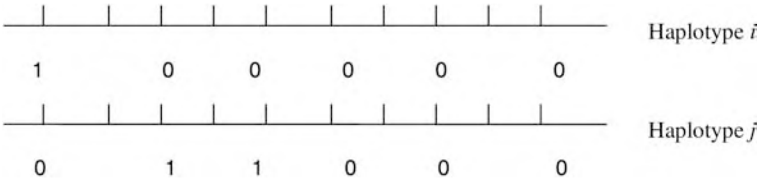


Figure 9 Scheme of length measure of similarity between haplotypes H_i and H_j.

large proportion of their sequence. To overcome such limitations, we need to develop other similarity measures.

4.1.2 Counting Measure

The counting measure $S(H_i, H_j)$ is defined as the number of alleles in common between haplotypes H_i and H_j. The haplotypes may not be identical, but share some common alleles. If the shared common alleles contain disease alleles, all haplotypes sharing disease alleles may be present more often in the affected individuals than the expected.

4.1.3 Length Measure

The length measure $S(H_i, H_j)$ is defined as the length of the longest continuous interval of matching alleles between haplotypes H_i and H_j. As shown in Figure 9, the length measure between haplotypes H_i and H_j is 7.

The length measure characterizes partial sharing identical by descent (IBD) between haplotypes.

4.2 Test Statistics

There are two types of test statistics: statistics based on the similarity measure of all haplotypes and statistics based on the similarity measure of part of the haplotypes. For the convenience of presentation, the former will be referred to as the overall similarity measure, the latter as the partial similarity measure.

We first consider test statistic based on the overall similarity measure. Define $\Sigma = \mathrm{diag}(P_1, \ldots, P_m) - PP^{\mathrm{T}}$ and $b = 2SP$. Then, $\mathrm{var}(\Gamma)$ and $\mathrm{var}(\Gamma^A)$ can be approximated by $\mathrm{var}(\Gamma) = \frac{4}{n_G} P^{\mathrm{T}} S^{\mathrm{T}} \Sigma S P$ and $\mathrm{var}(\Gamma)^A = \frac{4}{n_A} (P^A)^{\mathrm{T}} (S^A)^{\mathrm{T}} \Sigma^A S^A P^A$, where $\Sigma^A = \mathrm{diag}(P_1^A, \ldots, P_m^A) - P^A (P^A)^{\mathrm{T}}$, S^A and P^A are defined as before. We define statistic based on the overall haplotype similarity measure as:

$$T_{\mathrm{os}} = \frac{(\hat{\Gamma}^A - \hat{\Gamma})^2}{4\left[\frac{1}{n_G} \hat{P}^{\mathrm{T}} \hat{S}^{\mathrm{T}} \hat{\Sigma} \hat{S} \hat{P} + \frac{1}{n_A} (\hat{P}^A)^{\mathrm{T}} (\hat{S}^A)^{\mathrm{T}} \hat{\Sigma}^A \hat{S}^A \hat{P}^A\right]}$$

under the null hypothesis of no association, T_{os} is asymptotically distributed as a central $\chi^2_{(1)}$ distribution.

Now consider the test based on the partial haplotype similarity measure. Let $b_{ii} = S_{ii}P_{H_i} + \sum_{j=1}^{m} S_{ij}P_{H_j}$ and $b_{ij} = P_{H_i}S_{ij}P_{H_j}$ $(i \neq j)$. $\Gamma_H = [\Gamma_{H_1}, \ldots, \Gamma_{H_m}]^{\mathrm{T}}$ and $B = (b_{ij})_{m\times m}$. B^A and Γ_H^A for the affected individuals are similarly defined. Define the test statistic as $T_s = (\Gamma_H^A - \Gamma_H)^{\mathrm{T}}\Lambda^-(\Gamma_H^A - \Gamma_H)$, where $\Lambda = \frac{1}{n_G}B\Sigma B^{\mathrm{T}} + \frac{1}{n_A}B^A\Sigma^A(B^A)^{\mathrm{T}}$. Under the null hypothesis of no association, T_s is asymptotically distributed as a central $\chi^2_{(r)}$ distribution, where $r = \mathrm{rank}(\Lambda)$.

Above we also show that the similarity measure of the genome region is a quadratic function of the allele or haplotype frequencies. Therefore, similarity-based statistics are nonlinear test statistics.

5 Generalized T^2 Test Statistic

Although most researchers acknowledge that genetic variation provides valuable information for diagnosis, prevention and treatment of complex diseases, there is no universally accepted consensus on how genetic variation contributes to the cause of complex diseases [90, 91]. Two different basic view points on the genetic architecture of complex diseases lead to two different strategies for analyzing complex diseases.

The popular view on the mechanisms of the common diseases is to assume that a single marker or gene acts independently and can explain the pathogenesis of the disease. The widely used strategies for unraveling the genetic structure of a common disease are single-locus analysis, one locus at a time, assuming that each susceptible locus has a large independent main effect on disease risk. The strategies focusing on the single genes with large marginal effects on disease risk have resulted in only limited success in genetic studies of complex diseases [47]. Current research demonstrates that only a small proportion of the disease risk is due to the influences of variations in a single gene with large genetic effects. In the real world, complex diseases develop as a consequence of interactions between multiple DNA variants, and exposure to environmental agents varying over time and space, which are organized into networks. The single-gene paradigm for genetic analysis which has proven successful in dissecting genetic structures of Mendelian diseases may not lead to success in genetic studies of complex diseases.

There are an increasing number of researchers who advocate taking a systems-level approach to complex diseases. The new concept concerning complex disease is to assume that the development of disease should be considered as a dynamic process with gene–gene and gene–environment interactions contributing within a complex biological system which is hierarchically organized into complex gene interaction networks [90, 91]. Genetic mechanisms underlying complex diseases involve multiple genes which influence disease variation largely through genetic interaction networks in which the effect of one gene is enhanced or masked by effects of other genes [66]. Actions of genes through networks are consequences of complex molecular interactions occurring during biological processes such as metabolism, transcription, signal transduction and translations. It was reported that the principles underlying stable phenotypes in the presence of mutations comprise genetic network structure that are redundant and robust [39]. Consequently, the genetic effects on the phenotype can be observed only when multiple mutations hit the genetic networks. The phenotypic variations depend on the genetic interaction networks. Uncovering hierarchically organized gene interaction networks and their nonadditive relationships with disease susceptibility require developing novel statistics which can test association of a set of markers or networks with the disease.

The Hotelling T^2 statistic can be used to test association of multiple markers (or networks) with the disease [117].

5.1 Test Statistic

Consider a design in which n_A cases from an affected population and n_G control subjects from a comparable unaffected population are sampled. Suppose that there are k markers typed in the samples. The j-th marker has alleles B_j and b_j with population frequencies P_{B_j} and P_{b_j}, respectively. Define an indicator variable for the genotype of the j-th marker for the i-th individual from the affected population:

$$X_{ij} = \begin{cases} 1 & B_jB_j \\ 0 & B_jb_j \\ -1 & b_jb_j \end{cases} .$$

Similarly, we define an indicator variable, Y_{ij}, for an individual from the unaffected population. Let:

$$\begin{aligned} X_i &= (X_{i_1}, \ldots, X_{i_k})^{\mathrm{T}},\ Y_i = (Y_{i_1}, \ldots, Y_{i_k})^{\mathrm{T}} \\ \bar{X}_j &= \frac{1}{n_A}\sum_{i=1}^{n_A} X_{ij},\ \bar{Y}_j = \frac{1}{n_G}\sum_{i=1}^{n_G} Y_{ij} \\ \bar{X} &= (\bar{X}_1, \ldots, \bar{X}_k)^{\mathrm{T}},\ \bar{Y} = (\bar{Y}_1, \ldots, \bar{Y}_k)^{\mathrm{T}} . \end{aligned}$$

The pooled-sample variance-covariance matrix of the indicator variables for the marker genotypes is defined as:

$$S = \frac{1}{n_A + n_G - 2}\left[\sum_{i=1}^{n_A}(X_i - \bar{X})(X_i - \bar{X})^{\mathrm{T}} + \sum_{i=1}^{n_G}(Y_i - \bar{Y})(Y_i - \bar{Y})^{\mathrm{T}}\right] .$$

Hotelling's [51] T^2 statistic is then defined as

$$T^2 = \frac{1}{\frac{1}{n_A} + \frac{1}{n_G}}(\bar{X} - \bar{Y})^{\mathrm{T}} S^{-1}(\bar{X} - \bar{Y}) ,$$

Under the null hypothesis of no association of markers with the disease, the statistic T^2 is asymptotically distributed as a central $\chi^2_{(k)}$ distribution.

5.2 Nonlinear T^2 test

We can also develop a nonlinear T^2 statistic for testing the association of a set of markers with the disease. Let f be a nonlinear function. Then, $f(\bar{Y}) - f(\bar{X})$ is asymptotically distributed as the following multivariate normal distribution:

$$f(\bar{Y}) - f(\bar{X}) \sim N(f(\mu_Y) - f(\mu_X), \Lambda),$$

$$\mu_x = E[\bar{X}] ,$$

$$\mu_Y = E[\bar{Y}],\ \Lambda\frac{1}{n_A}C\Sigma_Y C^T + \frac{1}{n_G}B\Sigma_x B^{\mathrm{T}},\ C = \left(\frac{\partial f}{\partial \mu_Y^T}\right),$$

$$\Sigma_Y = n_A \mathrm{cov}(\bar{Y}, \bar{Y}),\ \mu_x = E[\bar{X}] ,$$

$$B = \left(\frac{\partial f}{\partial \mu_x}^{\mathrm{T}}\right),\ \Sigma_x = n_G \mathrm{cov}(\bar{X}, \bar{X}) .$$

The nonlinear T^2 statistic can be defined as:

$$T_{\mathrm{N}} = [f(\bar{Y}) - f(\bar{X})]^{\mathrm{T}}\,(\hat{\Lambda})^{-}\,[f(\bar{Y} - f(\bar{X})]$$

where $(\hat{\Lambda})^{-}$ is the generalized inverse of the matrix $\hat{\Lambda}$. Let $r = \mathrm{rank}(\hat{\Lambda})$. It can be shown [43] that under the null hypothesis of no association of the markers with the disease, the statistic T_{N} is asymptotically distributed as a central $\chi^2_{(r)}$ distribution.

6 Family-based Association Studies

Although population-based association studies are a powerful tool for genetic studies of complex diseases, their drawback is that they may create spurious association due to population substructure. Recently, genomic control has been proposed to adjust for the effect of population substructure [27, 29]. However, the adjustment depends on the assumed models and the population parameters [124]. Since the true disease models are unknown and the estimation of population parameters may not be accurate, genomic control cannot completely eliminate the effect of population substructure on association studies.

The transmission/disequilibrium test (TDT), introduced by Spielman and coworkers [94], and its extensions are widely used to test for linkage in the presence of association because they can avoid spurious association due to population substructure [34]. The original TDT and its early extensions use one marker at a time [59, 63, 84, 89, 93]. With availability of a large collection of SNPs and the rapid progress of efficient genotyping methods, the TDT has been extended to using haplotypes and multiple marker loci [10, 11, 18, 65, 77, 121, 122]. Although collection of nuclear family data is not easy for late onset diseases, TDT will still be a major tool for genome-wide association studies.

6.1 TDT at a Single Locus with Two Alleles

Throughout the chapter, we consider nuclear families with at least one affected child. For the time being, we consider a locus with two alleles B_1 and B_2 at the marker locus and assume that the parental genotypes are known. We also assume that parents are heterozygous at the marker locus. Let n_1 be the total number of transmissions of allele B_1 to affected children and $n_2$be the total number of transmissions of allele B_2. Then, the original TDT is defined as [94]:

$$\mathrm{TDT} = \frac{(n_1 - n_2)^2}{n_1 + n_2} . \tag{4}$$

The original TDT can be rewritten in terms of allele frequencies. Let $n = n_1 + n_2$, $\hat{p} = n_1/n$, and $\hat{q} = n_2/n$ be the frequencies of the transmitted alleles B_1 and B_2, respectively. Then, the TDT in Eq. (4) can be rewritten as:

$$\mathrm{TDT} = n(\hat{p} - \hat{q})^2 .$$

6.2 TDT at a Single Locus with Multiple Alleles or at Multiple Loci with Phase-known Haplotypes

The two-allele TDT can be generalized to multiple alleles or multiple loci with haplotypes. If the phases of haplotypes are known, the TDT formulations for multiple alleles or haplotypes are the same. For simplicity of presentation, we only consider haplotypes. However, conclusions obtained for haplotypes hold for multiple alleles. Assume that the number of haplotypes is k. Let n_{ij} be the number of times that heterozygous parents with genotype H_iH_j transmit haplotype H_i to an affected child. Since we consider only heterozygous parents, we assume $n_{ii} = 0$ for all i. Let $n_{i\cdot} = \sum_{j=1}^{k} n_{ij}$ be the number of times that haplotype H_i is transmitted to an affected child and $n_{\cdot i} = \sum_{j=1}^{k} n_{ji}$ be the number of times that it is not. The extension of the two-allele TDT is then given by [93]:

$$\mathrm{TDT}_m = \frac{k-1}{k} \sum_{i=1}^{k} \frac{(n_{i\cdot} - n_{\cdot i})^2}{n_{i\cdot} + n_{\cdot i}} \ . \tag{5}$$

This test statistic may not be distributed as a $\chi^2_{(k-1)}$ distribution [59,84,88].

To ensure that the multiple allele or haplotype TDT asymptotically follows a χ^2 distribution, TDT_m can be modified using score method [88]. Let $n = \sum_{i=1}^{k} \sum_{j=1}^{k} n_{ij}$. Let p_{ij} be the probability that a parent with genotype $H_i H_j$ transmits haplotype H_i to an affected child, $p_{i\cdot}$ be the probability that haplotype H_i is transmitted to an affected child and $p_{\cdot i}$ be the probability that haplotype H_i is not transmitted to an affected child. Their estimates are, respectively, given by:

$$\hat{p}_{ij} = \frac{n_{ij}}{n} \ , \quad \hat{p}_{i\cdot} = \frac{n_{i\cdot}}{n} \quad \text{and} \quad \hat{p}_{\cdot i} = \frac{n_{\cdot i}}{n} \ .$$

Let $\hat{p} = [\hat{p}_{1\cdot}, \ldots, \hat{p}_{k\cdot}]^{\mathrm{T}}$, $\hat{q} = [\hat{p}_{\cdot 1}, \ldots, \hat{p}_{\cdot k}]^{\mathrm{T}}$, $\hat{q} = [\hat{p}_{\cdot 1}, \ldots, \hat{p}_{\cdot k}]^{\mathrm{T}}$ and $d = \hat{p} - \hat{q}$. We can show that the variance-covariance matrix of d is given by:

$$\mathrm{cov}(d, d) = \Sigma / n \ ,$$

where:

$$\Sigma = \Sigma_p + \Sigma_q - \Sigma_{pq} - \Sigma_{pq}^{\mathrm{T}} \ ,$$

$$\begin{aligned}
\Sigma_p &= \begin{bmatrix} p_{1\cdot}(1 - p_{1\cdot}) & \ldots & -p_{1\cdot} p_{k\cdot} \\ \ldots & \ldots & \ldots \\ -p_{k\cdot} p_{1\cdot} & \ldots & p_{k\cdot}(1 - p_{k\cdot}) \end{bmatrix} \\
\Sigma_q &= \begin{bmatrix} p_{1\cdot}(1 - p_{\cdot 1}) & \ldots & -p_{\cdot 1} p_{\cdot k} \\ \ldots & \ldots & \ldots \\ -p_{\cdot k} p_{\cdot 1} & \ldots & p_{\cdot k}(1 - p_{\cdot k}) \end{bmatrix} \\
\Sigma_{pq} &= \begin{bmatrix} -p_{1\cdot} p_{\cdot 1} & \ldots & p_{1k} - p_{1\cdot} p_{\cdot k} \\ \ldots & \ldots & \ldots \\ p_{k1} - p_{k\cdot} p_{\cdot 1} & \ldots & -p_{k\cdot} p_{\cdot k} \end{bmatrix} \ .
\end{aligned} \tag{6}$$

By large sample theory [60], d is asymptotically as $N(p - q, \frac{1}{n}\Sigma)$.

Define the statistic:

$$\mathrm{TDT}_s = n d^{\mathrm{T}} \Sigma^{-} d \ , \tag{7}$$

where Σ^- is the generalized inverse of matrix Σ. Then, under the null hypothesis of no linkage, TDT_s is asymptotically distributed as a central χ^2 distribution with degrees of freedom r, where $r = \text{rank}(\Sigma^-)$.

6.3 Sib-TDT

6.3.1 Comparison of Genotype Frequencies

Assume that total N_s sibships are sampled. Let G be the number of observed genotypes among all individuals in the sample. Let m_{ij} be the number of individuals with the j-th genotype in the i-th sibship. Let N_i^A and N_i^C be the number of affected individuals and unaffected individuals in the i-th sibship, respectively. We denote the total number of individuals in the i-th sibship by N_i, i.e. $N_i = N_i^A + N_i^C$. Let t_{ij} be the number of individuals with the j-th genotype in the i-th sibship. Then, m_{ij} follows a hypergeometric distribution with mean:

$$e_{ij} = t_{ij} \frac{N_i^A}{N_i} ,$$

and variance:

$$V_{ij} = t_{ij} \frac{N_i^A}{N_i} \left(1 - \frac{N_i^A}{N_i}\right) \frac{N_i - t_{ij}}{N_i - 1} .$$

Let $\pi_{ij} = \frac{t_{ij}}{N_i}$, $\pi_i = [\pi_{i1}, \ldots, \pi_{iG}]^{\mathrm{T}}$, $t_i = [t_{i1}, \ldots, t_{iG}]^{\mathrm{T}}$, $e_i = [e_{i1}, \ldots, e_{iG}]^{\mathrm{T}}$, $m_i = [m_{i1}, \ldots, m_{iG}]^{\mathrm{T}}$ and $V_i = [\text{diag}(\pi_i) - \pi_i \pi_i^{\mathrm{T}}] \frac{N_i^A N_i^C}{N_i - 1}$.

The total observed genotype counts for the affected sibs across all families with its expected value under the null hypothesis of no association are given by:

$$m = \sum_{i=1}^{N_s} m_i$$

and:

$$e = \sum_{i=1}^{N_s} e_i .$$

Let N^A be the total number of affected sibs across all families. Let $P = \frac{m}{N^A}$, $\mu = \frac{e}{N^A}$ and $V = \frac{1}{(N^A)^2} \sum_{i=1}^{N_s} V_i$. When the sample size is large, P is asymptotically distributed as a multivariate normal distribution with mean μ and variance-covariance matrix V. Let $b_{jj} = -1 - \log P_j$, $b_{jk} = 0$, $j \neq k$, $B = (b_{jk})_{G \times G}$, $Y_j = -P_j \log P_j$, $Y = [Y_1, \ldots, Y_G]^{\mathrm{T}}$ and $\Sigma = BVB^{\mathrm{T}}$.

Define the test statistic:

$$T_{\mathrm{SG}} = (Y - \mu)^{\mathrm{T}} \Sigma^{-} (Y - \mu) ,$$

where Σ^{-} denotes the generalized inverse of matrix Σ. Under the null hypothesis of no association, statistic T_{SG} is asymptotically distributed as a $\chi^2_{(G-1)}$.

6.3.2 Comparison of Allele Frequencies

When the number of genotypes is large, an approach via the comparison of genotype frequencies will have low power. To avoid this problem, we compare allele frequencies. Allele counts are linear combinations of genotype counts [85]. Suppose that there are K alleles. The count of the l-th allele for the affected individuals in the i-th sibship is equal to $a_{il} = C_{il}^{\mathrm{T}} m_i$, where the j-th element of the vector C_{il} has the value 2, 1 or 0, depending on whether the corresponding genotype has 2, 1 or 0 alleles of type$l$ allele. Let $C_i^{\mathrm{T}} = [C_{i1}, \ldots, C_{ik}]$. Then, the observed allele count of the affected individuals in the i-th sibship and its expected values under the assumption of no association are given by $a_i = C_i m_i$ and $e_i^a = C_i e_i$. The covariance matrix is $V_i^a = C_i V_i C_i^{\mathrm{T}}$.

Define $m^a = \sum_{i=1}^{N_s} a_i$, $e^a = \sum_{i=1}^{N_s}$, and $V^a = \sum_{i=1}^{N_s} V_i^a$. Let $P^a = \frac{m^a}{N^A}$, $\mu^a = \frac{e^a}{N^A}$ and $\Sigma^a = \frac{BV^aB^{\mathrm{T}}}{(N^A)^2}$, where B is defined as that in the previous section.

Define $Y_j^a = -P_j^a \log P_j^a$ and $Y^a = [Y_1^a, \ldots, Y_k^a]^{\mathrm{T}}$. Then, the test statistic T_{Sa}, defined as:

$$T_{\mathrm{Sa}} = (Y^a - \mu^a)^{\mathrm{T}} (\Sigma^a)^{-} (Y^a - \mu^a) ,$$

is asymptotically distributed as a central $\chi^2_{(k-1)}$ under the null hypothesis of no association.

7 Nonlinear Transmission/Disequilibrium Test

Similar to population-based association studies, multiple-testing problems are also the major barrier to success of genome-wide TDT. By the same arguments as those for population-based genome-wide association studies, there is an urgent need to develop novel genome-wide TDT statistics with high power.

A natural way of developing powerful genome-wide TDT statistics is to apply a nonlinear transformation of the number of the transmitted and nontransmitted alleles or haplotypes for enlarging the difference in the number of transmitted and nontransmitted alleles or haplotypes. The common feature of the original TDT and its extensions is to compare the difference between the numbers of alleles or haplotypes transmitted and not transmitted from heterozygous parents to affected children, respectively. Therefore, amplifying the difference between the numbers of transmitted and nontransmitted alleles

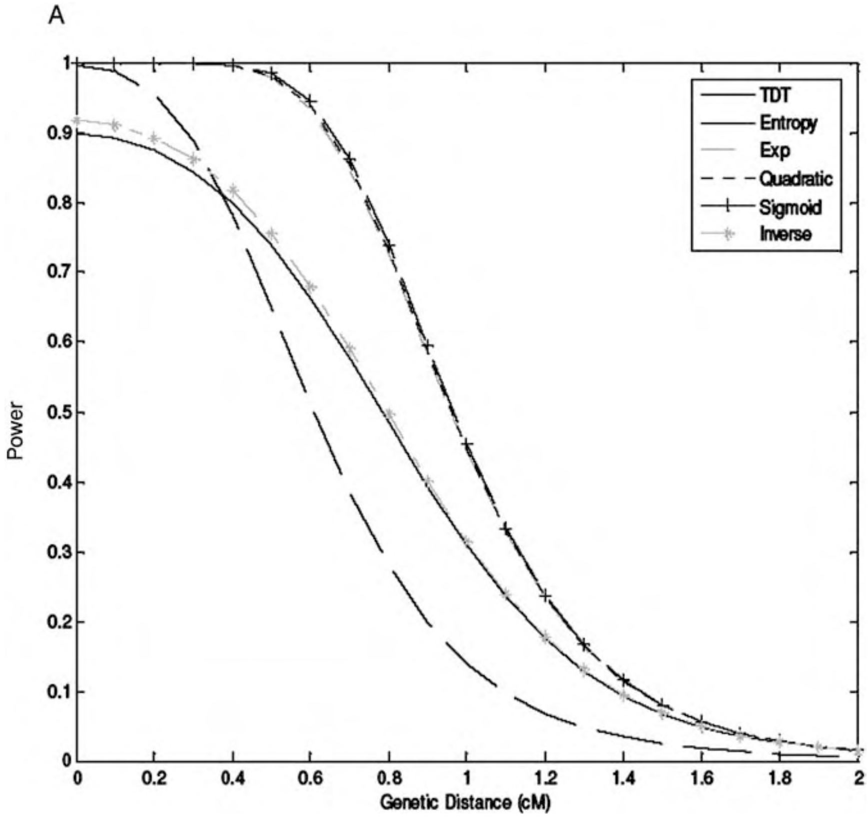


Figure 10 Power of nonlinear test statistics as a function of genetic distance.

or haplotypes is essential to improving the power of the TDT statistic. We hypothesized that this goal can be achieved by modifying the TDT statistics. Instead of comparing the difference between the numbers of the transmitted and nontransmitted alleles or haplotypes, the modified TDT compares the difference between a nonlinear transformation of these numbers. Such a transformation should satisfy two conditions: (i) statistics based on such a nonlinear transformation must still provide valid tests in the presence of population substructure and (ii) the nonlinear transformation should amplify the difference between the numbers of transmitted and nontransmitted alleles or haplotypes. The TDT based on the comparison of differences between the nonlinear transformations of the numbers of transmitted and nontransmitted alleles or haplotypes is referred to as nonlinear TDT.

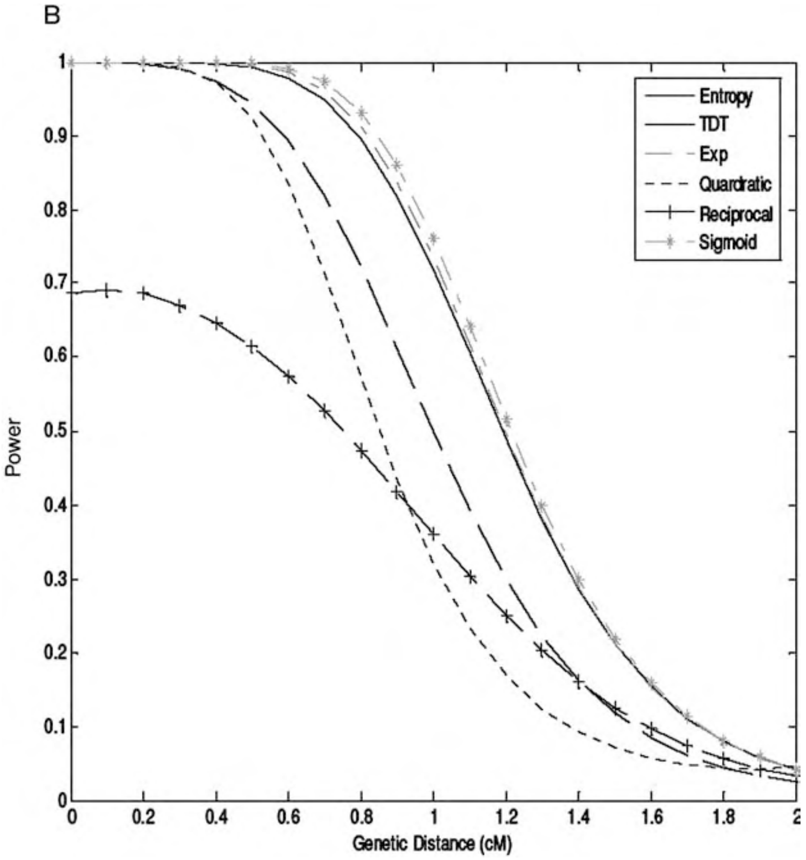


Figure 10 (continued)

7.1 General Procedures for the Construction of the Nonlinear TDT

For the simplicity, we consider family trios (one affected offspring with two heterozygous parents) and assume that all family members are genotyped at one or multiple marker loci. For haplotype analysis, we assume that the phases of the haplotypes are known. The results can be extended to sib-pair and general pedigrees with phase known and unknown haplotypes£®

7.1.1 A Single Locus with Two Alleles

Consider a locus with two alleles M_1 and M_2. Let n_1 be the total number of transmission of allele M_1 to the affected child and n_2 be the total number of transmission of allele M_2. Let $n = n_1 + n_2$, $\hat{p} = n_1/n$ and $\hat{q} = n_2/n$ be the estimated frequencies of the transmitted alleles M_1 and M_2, respectively. The nonlinear TDT for one locus with two alleles is defined as:

$$\mathrm{TDT}_{N_1} = \frac{n[f(\hat{p}) - f(\hat{q})]^2}{\hat{p}\hat{q}\{[f'(\hat{p}) + f'(\hat{q})]^2\}}\,, \tag{8}$$

where $f(\cdot)$ is a differentiable nonlinear function satisfying some regularity conditions such as entropy, exponential, polynomial, sigmoid and reciprocal functions. It can be shown that TDT_{N_1} is asymptotically distributed as a central $\chi^2_{(1)}$ distribution under the null hypothesis of no linkage or no association and asymptotically distributed as a noncentral $\chi^2_{(1)}$ distribution with the noncentrality parameter:

$$\lambda_{\mathrm{N}_1} = \frac{n[f(p) - f(q)]^2}{pq\{[f'(p) + f'(q)]^2\}}\,, \tag{9}$$

under the alternative hypothesis of presence of linkage and association.

7.1.2 **A Single Locus with Multiple Alleles or Multiple Loci with Phase-known Haplotypes**

The two-allele TDT can be generalized to multiple alleles or multiple loci with haplotypes. If phases of haplotypes are known, the TDT formulations for multiple alleles or haplotypes are the same. For simplicity of presentation, we only consider haplotypes. However, conclusions obtained for haplotypes also hold for multiple alleles. Assume that the number of haplotypes is K. Let n_{ij} be the number of times that heterozygous parent with genotype H_iH_j transmits H_i to an affected child. Since we consider only heterozygous parents, we assume $n_{ii} = 0$ for all i. Let $n_{i\cdot} = \sum_{j=1}^{k} n_{ij}$ be the number of times that haplotype H_i is transmitted to an affected child and $n_{\cdot i} = \sum_{j=1}^{k} n_{ji}$ be the number of times that is not. Furthermore, let $n = \sum_{i=1}^{k}\sum_{j=1}^{k} n_{ij}$. Let P_{ij} be the probability that a parent with genotype H_iH_j transmits haplotype H_i to an affected child, $P_{i\cdot}$ be the probability that haplotype H_i is transmitted to an affected child and $P_{\cdot i}$ be the probability that haplotype H_i is not transmitted to an affected child. Their estimates are given by $\hat{p}_{ij} = n_{ij}/n$, $\hat{p}_{i\cdot} = n_{i\cdot}/n$ and $\hat{p}_{\cdot i} = n_{\cdot i}/n$. Let:

$$\begin{aligned}
\Sigma_p &= \begin{bmatrix} p_{1\cdot}(1 - p_{1\cdot}), & \ldots, & -p_{1\cdot}p_{k\cdot} \\ \ldots & \ldots & \ldots \\ -p_{k\cdot}p_{1\cdot}, & \ldots, & -p_{k\cdot}(1 - p_{k\cdot}) \end{bmatrix} \\
\Sigma_q &= \begin{bmatrix} p_{\cdot 1}(1 - p_{\cdot 1}), & \ldots, & -p_{\cdot 1}p_{\cdot k} \\ \ldots & \ldots & \ldots \\ -p_{\cdot k}p_{\cdot 1}, & \ldots, & -p_{\cdot k}(1 - p_{\cdot k}) \end{bmatrix} \\
\Sigma_{pq} &= \begin{bmatrix} p_{11} - p_{1\cdot}p_{\cdot 1}, & \ldots, & p_{1k} - p_{1\cdot}p_{\cdot k} \\ \ldots & \ldots & \ldots \\ p_{k1} - p_{k\cdot}p_{\cdot 1}, & \ldots, & p_{kk} - p_{k\cdot}p_{\cdot k} \end{bmatrix}
\end{aligned}$$

$$\Sigma \quad = \quad \Sigma_p + \Sigma_q - \Sigma_{pq} - \Sigma_{pq}^{\mathrm{T}} \, . \tag{10}$$

Their corresponding estimates are denoted by $\hat{\Sigma}_p$, $\hat{\Sigma}_q$, $\hat{\Sigma}_{pq}$ and $\hat{\Sigma}$.

Let the vectors of the nonlinear transformation of the frequencies of the transmitted and nontransmitted haplotypes be $X = [f(\hat{p}_{1\cdot}), \ldots, f(\hat{p}_{k\cdot})]^{\mathrm{T}}$, $Y = [f(\hat{p}_{\cdot 1}), \ldots, f(\hat{p}_{\cdot k})]^{\mathrm{T}}$, respectively. Let B and C be the Jacobian matrices of X and Y, respectively, and define $b_{ii} = f'(\hat{p}_{i\cdot})$, $b_{ij} = 0$ for $i \neq j$, $B = (b_{ij})_{k \times k}$, $c_{ii} = f'(\hat{p}_{\cdot i})$, $c_{ij} = 0$ for $j \neq k$, $C = (c_{jk})_{k \times k}$. Let $\hat{B}$ and $\hat{C}$ be the estimates of matrices B and C, respectively. Let $\Lambda = B\Sigma_X B^{\mathrm{T}} + C\Sigma_Y C^{\mathrm{T}} - C\Sigma_{XY}B^{\mathrm{T}} - B\Sigma_{XY}C^{\mathrm{T}}$ and its estimates by $\hat{\Lambda}$. Then, the nonlinear TDT for haplotypes is defined as:

$$\mathrm{TDT}_{\mathrm{N}_2} = n(X - Y)^{\mathrm{T}} \hat{\Lambda}^{-} (X - Y) \, , \tag{11}$$

where $\hat{\Lambda}$ denotes a generalized inverse of the matrix Λ.

It can be shown that $\mathrm{TDT}_{\mathrm{N}_2}$ is asymptotically distributed as a central $\chi^2_{(r)}$ distribution under the null hypothesis of no linkage or no association and $r = \mathrm{rank}(\Lambda^{-})$. Under the alternative hypothesis that linkage and association between the marker and the disease loci exist, the statistic $\mathrm{TDT}_{\mathrm{N}_2}$ is asymptotically distributed as a noncentral $\chi^2_{(r)}$ distribution with the following noncentrality parameter:

$$\lambda_{\mathrm{N}_2} = n(\mu_{\mathrm{T}} - \mu_{\mathrm{NT}})^{\mathrm{T}} \Lambda^{-} (\mu_{\mathrm{T}} - \mu_{\mathrm{NT}}) \, , \tag{12}$$

where $\mu_{\mathrm{T}} = [f(p_{1\cdot}), \ldots, f(p_{m\cdot})]^{\mathrm{T}}$ and $\mu_{\mathrm{NT}} = [f(p_{\cdot 1}), \ldots, f(p_{\cdot k})]^{\mathrm{T}}$.

7.2 Power of the $N\backslash$ nonlinear TDT

A key component of power calculation is to compute noncentrality parameters. The noncentrality parameters of the χ^2 distribution of the nonlinear test statistics are formulated in Eqs. (10) and (13), which implies that the expected frequencies of the transmitted and nontransmitted alleles/haplotypes need to be calculated for power calculations.

We first study how to calculate the expected frequencies of the transmitted and nontransmitted alleles. It was shown that:

$$\begin{aligned} p_{i\cdot} \quad &= \quad p_i + b(1-\theta)\delta_{1i} \quad \text{and} \quad p_{\cdot i} = p_i + b\theta\delta_{1i} \, , \\ \text{where} \quad b \quad &= \quad \frac{(f_{11} - f_{12})P_D + (f_{12} - f_{22})P_d}{P(A)} \, , \end{aligned}$$

P_D and P_d and are the frequencies of the alleles D and d at the disease locus, respectively. f_{11}, f_{12} and f_{22} are the penetrance for genotypes *DD, Dd* and *dd*, respectively, with $f_{11} \geq f_{12} \geq f_{22} \geq 0$, $P(A) = f_{11}P_D^2 + 2f_{12}P_D P_d + f_{22}P_d^2$ represents the disease prevalence in the studied population. θ is the recombination fraction between the marker and the disease locus, M_i denotes a marker allele, p_i is the frequency of the marker allele M_i and P_{D_i} is the frequency of the haplotype DM_i, δ_{1i} is the measure of LD between the marker allele M_i and allele at the disease locus and is defined as $\delta_{1i} = P_{D_i} - P_D P_i$. Thus, the expected frequencies of the transmitted and nontransmitted alleles are given by $E[P_{i\cdot}] = P_i + (1-\theta)E(\delta_{1i}]$ and $E[P_{\cdot i}] = P_i + b\theta E[\delta_{1i}]$, where $E[\delta_{1i}] = \delta_{1i}(0)(1-\theta)^t$, $\delta_{1i}(0)$ is the measure of the initial LD when the LD was created, t is the time since the generations of the LD between the marker and the disease locus. Next, we calculate the expected frequencies of the transmitted and nontransmitted haplotypes. For simplicity of notations, we consider only three-locus haplotypes. The extension of the methods for three-locus haplotypes to haplotypes with more than three loci is straightforward, but involves more complex notation. To provide guidance for study design and intuitively illustrate the validity of the nonlinear statistics for testing the null hypothesis, we approximate the noncentrality parameter. We can show that the noncentrality parameter of the nonlinear test statistics can be approximated by:

$$\begin{aligned}\lambda_{\mathrm{N}_2} \quad \approx \quad & n\delta_{\mathrm{DM}}^{\mathrm{T}}(1-2\theta)^2 b^2 \left(I + \frac{1}{2}S\right)^{\mathrm{T}} [(I+S)\Sigma_p(I+S)^{\mathrm{T}} - (I+S)\Sigma_{pq} \\ & - \Sigma_{pq}^{\mathrm{T}}(I+S)^{\mathrm{T}}\Sigma_p]^{-} \left(I + \frac{1}{2}S\right)\delta_{\mathrm{DM}} \,,\end{aligned}$$

where $\delta_{\mathrm{DM}} = [\delta_{11}, \dots, \delta_{1k}]^{\mathrm{T}}$, $S = b(1-2\theta)C^{-}H\delta_{\mathrm{DM}}$, C is a Jacobian matrix as defined in the text, $H = [H_1, \dots, H_k]^{\mathrm{T}}$ and $H_l = \mathrm{diag}(0, \dots, 0, f''(P_{\cdot l}), 0, \dots, 0)$. The quantities b, θ and the matrices Σ_p, Σ_{pq} are defined as before.

To ensure that the distributions of the nonlinear test statistics are a central χ^2 distribution, we need to have:

$$(1-2\theta)\delta_{1i} = 0 \quad \text{for all} \quad i = 1, \dots, k \,. \tag{13}$$

This demonstrates that either $\theta = 1/2$ or $\delta_{1i} = 0$ for all $i = 1, \dots, k$ will lead to Eq. (13). Therefore, like the original TDT, the nonlinear TDT statistics can also be used to test either linkage or association. To detect linkage by nonlinear TDT, the LD between at least one marker allele and a disease allele must exist, i.e. only in the presence of association, the nonlinear TDT can be used to test linkage. The most properties of the original TDT will also hold for the nonlinear TDT statistics. We conduct preliminary power studies of nonlinear TDT and plot the power curves of the standard and nonlinear TDT for the recessive and dominance disease models in Figures 10A and 10B, respectively.

7.3 Real Examples

To further evaluate their performance, the nonlinear TDT statistics were applied to two real data sets. The first data set is a test of association of the

RET gene, which encodes a receptor tyrosine kinase, with Hirschsprung [8]. The second example is to test the association of Fcγ receptor genes (FcγRIIA, FcγRIIIA and FcγRIIIB) with systemic lupus erythematosus in 126 pedigrees [32]. SNPs in these three genes were typed. The results are shown in Tables 2 and 3. The results showed that the p-values of most nonlinear TDT statistics are much smaller than those of the original TDT statistic. These results suggest that developing nonlinear TDT statistics harbors great potential for establishing genome-wide linkage or association.

Table 2 Test for association of the *RET* gene with HSCR

	TDT	Entropy	Exponential	Quadratic	Sigmoid	Reciprocal
Single SNP						
A45A	9.4E-8	$<$ E-16	9.4E-13	1.1E-13	4.5E-14	0.0086
V125V	0.034	9.1E-9	0.0022	0.0013	0.0011	0.0037
A432A	0.65	0.64	0.65	0.65	0.65	0.32
G691S	0.21	0.17	0.20	0.20	0.20	0.00011
L769L	0.11	0.080	0.10	0.10	0.10	0.063
S836S	0.48	0.43	0.47	0.47	0.46	0.021
S904S	0.21	0.17	0.20	0.20	0.20	0.0011
7-SNP Haplotypes						
A-L	6.2E-7	4.8E-9	1.1E-16	$<$ E-16	2.6E-15	0.0063

Table 3 Test for association of FcγR gene with SLE

	TDT	Entropy	Exponential	Quadratic	Sigmoid	Reciprocal
2-SNP Haplotypes	1.2E-5	5.8E-8	1.9E-7	2.4E-7	1.3E-7	0.0023

8 Perspective of Genome-wide Association Studies

Several large-scale genome-wide association studies in which 300 000 or 500 000 panels of SNPs will be typed are planned, including European initiatives, the National Institutes of Health (NIH) initiative for the Genetic Association Information Network, and the Genes and Environment Initiative [103]. To increase the power and reduce the cost of genotyping, two-stage study design for genome-wide association studies in which in the first stage all markers are typed in a fraction of samples and in the second stage a subset of markers showing significant association will be typed was proposed [86,110].

A key component of genome-wide association studies is to test interaction between genes and interaction between genes and environment. Complex diseases are caused by multiple genes, primarily through nonlinear gene interactions and gene–environment interactions. Complex gene interactions are organized into networks. Genetic interaction networks must be ubiquitous in common diseases, given the complex dynamic interactions of the genetic regulatory and metabolic networks. Despite growing consensus on the importance of testing for gene–gene interactions in genetic studies of complex diseases, the effect of gene–gene interactions has often been defined as a deviance from genetic additive effects, which is essentially treated as a residual term in genetic analysis and leads to low power to detect the presence of interacting effects. To what extent the definition of gene–gene interaction at population level reflects their biochemical or physiological interaction remains a mystery.

Testing of interactions including gene–gene and gene–environment interactions poses great challenge to genome-wide association studies because of the extremely large number of potential interactions. Should we incorporate testing of gene-gene interactions into genome-wide association studies? There is some doubt about the strategy to test for interactions at the expense of power for detecting main effects. Some suggest limiting the testing of interaction to pairs of markers that individually show significant association at some threshold. However, others argue that although both SNPs do not show significant evidence of association when analyzed individually, they do give significant evidence of interaction. It remains to be seen whether incorporating testing of interactions into genome-wide association studies will be useful.

We are entering a new era in genetic studies of complex diseases. Without question, the next few years will be an exciting time [122]. A dozen large-scale genome-wide association studies are under way. More and more powerful methods for genome-wide association studies are being developed. We can expect that a flurry of new association results will appear in the literature. The validation of association findings will continue to rely not only on the replication of independent samples, but also on functional studies of the SNPs. It is certain that genome-wide association studies, in conjunction with systems biology, will successfully dissect the complex structure of common diseases.

References

1 Ahmadi, K. R., M. E. Weale, Z. Y. Xue, et al. 2005. A single-nucleotide polymorphism tagging set for human drug metabolism and transport. Nat. Genet. **37**: 84–9.

2 Akey, J., L. Jin and M. Xiong. 2001. Haplotypes vs single marker linkage disequilibrium tests: what do we gain? Eur. J. Hum. Genet. **9**: 291–300.

3 Altshuler, D. and A. G. Clark. 2005. Genetics. Harvesting medical information from the human family tree. Science **307**: 1052–3.

4 Anderson, E. C. and J. Novembre. 2003. Finding haplotype block boundaries by using the minimum-description-length principle. Am. J. Hum. Genet. **73**: 336–54.

5 Bates, D. M. and D. G. Watts. 1980. Relative curvature measure of

nonlinearity. J. R. Stat. Soc. (Ser. B) **42**: 1–25.

6 BEAUMONT, M. A. AND B. RANNALA. 2004. The Bayesian revolution in genetics. Nat. Rev. Genet. **5**: 251–61.

7 BENJAMINI, Y. AND Y. HOCHBERG. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Stat. Soc. (Ser. B) **57**: 289–300.

8 BORREGO, S., A. RUIZ, M. E. SAEZ, et al. 2000. RET genotypes comprising specific haplotypes of polymorphic variants predispose to isolated Hirschsprung disease. J. Med. Genet. **37**: 572–8.

9 BORSTING, C., J. J. SANCHEZ AND N. MORLING. 2004. SNP typing on the NanoChip electronic microarray. Methods Mol. Biol. **297**: 155–68.

10 BOURGAIN, C., E. GENIN, P. MARGARITTE-JEANNIN AND F. CLERGET-DARPOUX. 2001. Maximum identity length contrast: a powerful method for susceptibility gene detection in isolated populations. Genet. Epidemiol. **21 (Suppl. 1)**: S560–4.

11 BOURGAIN, C., E. GENIN, C. OBER AND F. CLERGET-DARPOUX. 2002. Missing data in haplotype analysis: a study on the MILC method. Ann. Hum. Genet. **66**: 99–108.

12 BOURGAIN, C., E. GENIN, H. QUESNEVILLE AND F. CLERGET-DARPOUX. 2000. Search for multifactorial disease susceptibility genes in founder populations. Ann. Hum. Genet. **64**: 255–65.

13 CARDON, L. R. AND G. R. ABECASIS. 2003. Using haplotype blocks to map human complex trait loci. Trends Genet. **19**: 135–40.

14 CARLSON, C. S., M. A. EBERLE, L. KRUGLYAK AND D. A. NICKERSON. 2004. Mapping complex disease loci in whole-genome association studies. Nature **429**: 446–52.

15 CARLSON, C. S., M. A. EBERLE, M. J. RIEDER, J. D. SMITH, L. KRUGLYAK AND D. A. NICKERSON. 2003. Additional SNPs and linkage-disequilibrium analyses are necessary for whole-genome association studies in humans. Nat. Genet. **33**: 518–21.

16 CHAPMAN, N. H. AND E. M. WIJSMAN. 1998. Genome screens using linkage disequilibrium tests: optimal marker characteristics and feasibility. Am. J. Hum. Genet. **63**: 1872–85.

17 CLARK, C. W. 1990. *Mathematical Bioeconomics*. Wiley, New York, NY.

18 CLAYTON, D. 1999. A generalization of the transmission/disequilibrium test for uncertain-haplotype transmission. Am. J. Hum. Genet. **65**: 1170–7.

19 COLLINS, F. S. 2004. The case for a US prospective cohort study of genes and environment. Nature **429**: 475–7.

20 INTERNATIONAL HAPMAP CONSORTIUM. 2003. The International HapMap Project. Nature **426**: 789–96.

21 CORDELL, H. J. AND D. G. CLAYTON. 2005. Genetic association studies. Lancet **366**: 1121–31.

22 DALMASSO, C., P. BROET AND T. MOREAU. 2005. A simple procedure for estimating the false discovery rate. Bioinformatics **21**: 660–8.

23 DALY, E., J. D. RIUX, S. F. SCHAFFNER, T. J. HUDSON AND E. S. LANDER. 2001. High-resolution haplotype structure in the human genome. Nat. Genet. **29**: 229–32.

24 DAWSON, E., G. R. ABECASIS, S. BUMPSTEAD, et al. 2002. A first-generation linkage disequilibrium map of human chromosome 22. Nature **418**: 544–8.

25 DE VRIES, H. G., M. A. VAN DER MEULEN, R. ROZEN, D. J. HALLEY, H. SCHEFFER, L. P. TEN KATE, C. H. BUYS AND G. J. TE MEERMAN. 1996. Haplotype identity between individuals who share a CFTR mutation allele "identical by descent": demonstration of the usefulness of the haplotype-sharing concept for gene mapping in real populations. Hum. Genet. **98**: 304–9.

26 DEMPSTER, A. P., N. M. LAIRD AND R. D.B. 1977. Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. (Ser. B) **39**: 1–38.

27 DEVLIN, B. AND K. ROEDER. 1999. Genomic control for association studies. Biometrics **55**: 997–1004.

28 DEVLIN, B., K. ROEDER AND L. WASSERMAN. 2000. Genomic control for association studies: a semiparametric test to detect excess-haplotype sharing. Biostatistics **1**: 369–87.

29 DEVLIN, B., K. ROEDER AND L. WASSERMAN. 2001. Genomic control, a new approach to genetic-based association studies. Theor. Popul. Biol. **60**: 155–66.

30 DRAZEN, J. M., C. N. YANDAVA, L. DUBE, et al. 1999. Pharmacogenetic association between ALOX5 promoter genotype and the response to anti-asthma treatment. Nat. Genet. **22**: 168–70.

31 DRYSDALE, C. M., D. W. MCGRAW, C. B. STACK, et al. 2000. Complex promoter and coding region beta 2-adrenergic receptor haplotypes alter receptor expression and predict in vivo responsiveness. Proc. Natl Acad. Sci. USA **97**: 10483–8.

32 EDBERG, J. C., C. D. LANGEFELD, J. WU, et al. 2002. Genetic linkage and association of Fcγ receptor IIIA (CD16A) on chromosome 1q23 with human systemic lupus erythematosus. Arthritis Rheum. **46**: 2132–40.

33 EFRON, B. AND R. TIBSHIRANI. 2002. Empirical Bayes methods and false discovery rates for microarrays. Genet. Epidemiol. **23**: 70–86.

34 EWENS, W. J. AND R. S. SPIELMAN. 1995. The transmission/disequilibrium test: history, subdivision, and admixture. Am. J. Hum. Genet. **57**: 455–64.

35 EXCOFFIER, L. AND M. SLATKIN. 1995. Maximum-likelihood estimation of molecular haplotype frequencies in a diploid population. Mol. Biol. Evol. **12**: 921–7.

36 FAN, R. AND K. LANGE. 1998. Models for haplotype evolution in a nonstationary population. Theor. Popul. Biol. **53**: 184–98.

37 FREIMER, N. AND C. SABATTI. 2004. The use of pedigree, sib-pair and association studies of common diseases for genetic mapping and epidemiology. Nat. Genet. **36**: 1045–51.

38 GABRIEL, S. B., S. F. SCHAFFNER, H. NGUYEN, et al. 2002. The structure of haplotype blocks in the human genome. Science **296**: 2225–9.

39 GIBSON, G. AND G. WAGNER. 2000. Canalization in evolutionary genetics: a stabilizing theory? BioEssays **22**: 372–80.

40 GLAZIER, A. M., J. H. NADEAU AND T. J. AITMAN. 2002. Finding genes that underlie complex traits. Science **298**: 2345–9.

41 GOLDSTEIN, D. B. 2001. Islands of linkage disequilibrum. Nat. Genet. **29**: 109–111.

42 GOLDSTEIN, D. B., K. R. AHMADI, M. E. WEALE AND N. W. WOOD. 2003. Genome scans and candidate gene approaches in the study of common diseases and variable drug responses. Trends Genet. **19**: 615–22.

43 GREENWOOD, P. E. AND M. S. NIKULIN. 1996. *A Guide to Chi-squared Testing*. Wiley, New York, NY.

44 GUSFIELD, D. 2001. Inference of haplotypes from samples of diploid populations: complexity and algorithms. J. Comput. Biol. **8**: 305–23.

45 HALLDORSSON, B. V., V. BAFNA, R. LIPPERT, R. SCHWARTZ, F. M. DE LA VEGA, A. G. CLARK AND S. ISTRAIL. 2004. Optimal haplotype block-free selection of tagging SNPs for genome-wide association studies. Genome Res **14**: 1633–40.

46 HAMILTON, D. C. AND D. E. COLE. 2004. Standardizing a composite measure of linkage disequilibrium. Ann. Hum. Genet. **68**: 234–9.

47 HARTWELL, L. 2004. Genetics. Robust interactions. Science **303**: 774–5.

48 HELGASON, A., B. YNGVADOTTIR, B. HRAFNKELSSON, J. GULCHER AND K. STEFANSSON. 2005. An Icelandic example of the impact of population structure on association studies. Nat. Genet. **37**: 90–5.

49 HILL, W. G. AND A. ROBERTSON. 1968. Linkage disequilibrum in finite populations. Theor. Appl. Genet. **figs/38**: 226–31.

50 HORIKAWA, Y., N. ODA, N. J. COX, et al. 2000. Genetic variation in the gene encoding calpain-10 is associated with type 2 diabetes mellitus. Nat. Genet. **26**: 163–75.

51 HOTELLING, H. 1931. The generalization of Student's ratio. Ann. Math. Stat.**2**: 360–78.

52 JEFFREYS, A. J., L. KAUPPI AND R. NEUMANN. 2001. Intensely punctate meiotic recombination in the class II region of the major histocompatibility complex. Nat. Genet. **29**: 217–22.

53 JIMENEZ-SANCHEZ, G., B. CHILDS AND D. VALLE. 2001. Human disease genes. Nature **409**: 853–5.

54 JOHNSON, G. C., L. ESPOSITO, B. J. BARRATT, et al. 2001. Haplotype tagging for the identification of common disease genes. Nat. Genet. **29**: 233–7.

55 JORDE, L. B. 2000. Linkage disequilibrium and the search for complex disease genes. Genome Res **10**: 1435–44.

56 KE, X. AND L. R. CARDON. 2003. Efficient selective screening of haplotype tag SNPs. Bioinformatics **19**: 287–8.

57 KRAWEZAK, M., J. REISS AND D. N. COOPER. 1992. The mutational spectrum of single base-pair substitutions in mRNA splice junctions of human genes: causes and consequences. Hum. Genet.**90**: 41–54.

58 KRUGLYAK, L. AND D. A. NICKERSON. 2001. Variation is the spice of life. Nat. Genet. **27**: 234–6.

59 LAZZERONI, L. C. AND K. LANGE. 1998. A conditional inference framework for extending the transmission/disequilibrium test. Hum. Hered. **48**: 67–81.

60 LEHMANN, E. L. 1983. *Theory of Point Estimation*. Wiley, New York, NY.

61 LEWONTIN, R. C. 1964. The interaction of selection and linkage. II. Optimum models. Genetics **50**: 757–82.

62 LIN, S., D. J. CUTLER, M. E. ZWICK AND A. CHAKRAVARTI. 2002. Haplotype inference in random population samples. Am. J. Hum. Genet. **71**: 1129–37.

63 MARTIN, E. R., N. L. KAPLAN AND B. S. WEIR. 1997. Tests for linkage and association in nuclear families. Am. J. Hum. Genet. **61**: 439–48.

64 MARTIN, E. R., E. H. LAI, J. R. GILBERT, et al. 2000. SNPing away at complex diseases: analysis of single-nucleotide polymorphisms around APOE in Alzheimer disease. Am. J. Hum. Genet. **67**: 383–94.

65 MERRIMAN, T. R., I. A. EAVES, R. C. TWELLS, et al. 1998. Transmission of haplotypes of microsatellite markers rather than single marker alleles in the mapping of a putative type 1 diabetes susceptibility gene (IDDM6). Hum. Mol. Genet. **7**: 517–24.

66 MOORE, J. H. 2005. A global view of epistasis. Nat. Genet. **37**: 13–4.

67 MORRIS, R. W. AND N. L. KAPLAN. 2002. On the advantage of haplotype analysis in the presence of multiple disease susceptibility alleles. Genet. Epidemiol. **23**: 221–33.

68 NEALE, B. M. AND P. C. SHAM. 2004. The future of association studies: gene-based analysis and replication. Am. J. Hum. Genet. **75**: 353–62.

69 NIU, T. 2004. Algorithms for inferring haplotypes. Genet. Epidemiol. **27**: 334–47.

70 NIU, T., Z. S. QIN, X. XU AND J. S. LIU. 2002. Bayesian haplotype inference for multiple linked single-nucleotide polymorphisms. Am. J. Hum. Genet. **70**: 157–69.

71 NOTHNAGEL, M. 2002. Simulation of LD block-structured SNP haplotype data and its use for the analysis of case-control data by supervised learning methods. Am. J. Hum. Genet. **71 (Suppl.)**: A2363.

72 PATIL, N., A. J. BERNO, D. A. HINDS, et al. 2001. Blocks of limited haplotype diversity revealed by high-resolution scanning of human chromosome 21. Science **294**: 1719–23.

73 PONT-KINGDON, G., M. JAMA, C. MILLER, A. MILLSON AND E. LYON. 2004. Long-range (17.7 kb) allele-specific polymerase chain reaction method for direct haplotyping of R117H and IVS-8 mutations of the cystic fibrosis transmembrane regulator gene. J. Mol. Diagn. **6**: 264–70.

74 PRITCHARD, J. K. AND N. J. COX. 2002. The allelic architecture of human disease genes: common disease-common variant or not? Hum. Mol. Genet. **11**: 2417–23.

75 PRITCHARD, J. K. AND P. DONNELLY. 2001. Case-control studies of association in structured or admixed populations. Theor. Popul. Biol. **60**: 227–37.

76 QIN, Z. S., T. NIU AND J. S. LIU. 2002. Partition-ligation-expectation-maximization algorithm for haplotype inference with single-nucleotide polymorphisms. Am. J. Hum. Genet. **71**: 1242–7.

77 RABINOWITZ, D. AND N. LAIRD. 2000. A unified approach to adjusting association tests for population admixture with arbitrary pedigree structure and arbitrary missing marker information. Hum. Hered. **50**: 211–23.

78 RAO, D. C. 2001. Genetic dissection of complex traits: an overview. Adv. Genet. **42**: 13–34.

79 REICH, D. E., S. B. GABRIEL AND D. ALTSHULER. 2003. Quality and completeness of SNP databases. Nat. Genet. **33**: 457–8.

80 REICH, D. E. AND E. S. LANDER. 2001. On the allelic spectrum of human disease. Trends Genet. **17**: 502–10.

81 RICE, J. P., N. L. SACCONE AND E. RASMUSSEN. 2001. Definition of the phenotype. Adv. Genet. **42**: 69–76.

82 RISCH, N. AND K. MERIKANGAS. 1996. The future of genetic studies of complex human diseases. Science **273**: 1516–7.

83 SABATTI, C., S. SERVICE AND N. FREIMER. 2003. False discovery rate in linkage and association genome screens for complex disorders. Genetics **164**: 829–33.

84 SCHAID, D. J. 1996. General score tests for associations of genetic markers with disease using cases and their parents. Genet. Epidemiol. **13**: 423–49.

85 SCHAID, D. J. AND C. ROWLAND. 1998. Use of parents, sibs, and unrelated controls for detection of associations between genetic markers and disease. Am. J. Hum. Genet. **63**: 1492–506.

86 SCOL, A. D., L. J. SCOTT, G. R. ABECASIS AND M. BOEHNKE. 2006. Joint analysis is more efficient than replication-based analysis for two-stage genome-wide association studies. Nat. Genet. **figs/38**: 209–213.

87 SERFLING, R. J. 1980. *Approximation Theorems of Mathematical Statistics*. Wiley, New York, NY.

88 SHAM, P. C. 1997. Transmission/disequilibrum tests for multiallelic loci. Am. J. Hum. Genet. **61**: 774–8.

89 SHAM, P. C. AND D. CURTIS. 1995. An extended transmission/disequilibrium test (TDT) for multi-allele marker loci. Ann. Hum. Genet. **59**: 323–36.

90 SING, C. F., J. H. STENGARD AND S. L. KARDIA. 2004. Dynamic relationships between the genome and exposures to environments as causes of common human diseases. World Rev. Nutr. Diet **93**: 77–91.

91 SING, C. F., J. H. STENGARD AND S. L. KARDIA. 2003. Genes, environment, and cardiovascular disease. Arterioscler. Thromb. Vasc. Biol. **23**: 1190–6.

92 SMITH, D. J. AND A. J. LUSIS. 2002. The allelic structure of common disease. Hum. Mol. Genet. **11**: 2455–61.

93 SPIELMAN, R. S. AND W. J. EWENS. 1996. The TDT and other family-based tests for linkage disequilibrium and association. Am. J. Hum. Genet. **59**: 983–9.

94 SPIELMAN, R. S., R. E. MCGINNIS AND W. J. EWENS. 1993. Transmission test for linkage disequilibrium: the insulin gene region and insulin-dependent diabetes mellitus (IDDM). Am. J. Hum. Genet. **52**: 506–16.

95 STEPHENS, J. C., J. A. SCHNEIDER, D. A. TANGUAY, et al. 2001. Haplotype variation and linkage disequilibrium in 313 human genes. Science **293**: 489–93.

96 STEPHENS, M. AND P. DONNELLY. 2003. A comparison of Bayesian methods for haplotype reconstruction from population genotype data. Am. J. Hum. Genet. **73**: 1162–9.

97 STOREY, J. D. 2002. A direct approach to false discovery rates. J. R. Stat. Soc. (Ser. B) **64**: 479–98.

98 STOREY, J. D., J. E. TAYLOR AND D. SIEGMUND. 2004. Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: a unified approach. J. R. Stat. Soc. (Ser. B) **66**: 187–205.

99 STOREY, J. D. AND R. TIBSHIRANI. 2003. Statistical significance for genomewide studies. Proc. Natl Acad. Sci. USA **100**: 9440–5.

100 Strauch, K., R. Fimmers, M. P. Baur and T. F. Wienker. 2003. How to model a complex trait. 1. General considerations and suggestions. Hum. Hered. **55**: 202–10.

101 Stumpf, M. P. and D. B. Goldstein. 2003. Demography, recombination hotspot intensity, and the block structure of linkage disequilibrium. Curr. Biol. **13**: 1–8.

102 Szantai, E., A. Szilagyi, A. Guttman, M. Sasvari-Szekely and Z. Ronai. 2004. Genotyping and haplotyping of the dopamine D4 receptor gene by capillary electrophoresis. J. Chromatogr. A **1053**: 241–5.

103 Thomas, D. C. 2006. Are we ready for genome-wide association studies? Editorial. Cancer Epidemiol. Biomarkers Prev. **15**: 595–8.

104 Thomas, D. C. and D. G. Clayton. 2004. Betting odds and genetic associations. J. Natl Cancer Inst. **96**: 421–3.

105 Thomas, D. C., R. W. Haile and D. Duggan. 2005. Recent developments in genomewide association scans: a workshop summary and review. Am. J. Hum. Genet. **77**: 337–45.

106 Tzeng, J. Y., B. Devlin, L. Wasserman and K. Roeder. 2003. On the identification of disease mutations by the analysis of haplotype similarity and goodness of fit. Am. J. Hum. Genet. **72**: 891–902.

107 van den Oord, E. J. and B. M. Neale. 2004. Will haplotype maps be useful for finding genes? Mol. Psychiatry **9**: 227–36.

108 Van der Meulen, M. A. and G. J. Te Meerman. 1997. Haplotype sharing analysis in affected individuals from nuclear families with at least one affected offspring. Genet. Epidemiol. **14**: 915–20.

109 Wacholder, S., S. Chanock, M. Garcia-Closas, L. El Ghormli and N. Rothman. 2004. Assessing the probability that a positive report is false: an approach for molecular epidemiology studies. J. Natl Cancer Inst. **96**: 434–42.

110 Wang, H., D. C. Thomas, I. Pe'er and D. O. Stram. 2006. Optimal two-stage genotyping designs for genome-wide association scans. Genet. Epidemiol. **30**: 356–68.

111 Wang, W. Y., B. J. Barratt, D. G. Clayton and J. A. Todd. 2005. Genome-wide association studies: theoretical and practical concerns. Nat. Rev. Genet. **6**: 109–18.

112 Weir, B. S. 1979. Inferences about linkage disequilibrium. Biometrics **35**: 235–54.

113 Weir, B. S. and C. C. Cockerham. 1989. Complete characterization of disequilibrum at two loci. In Feldman, M. W. (ed.), *Mathematical Evolutionary Theory*. Princeton University Press, Princeton, NJ: 86–110.

114 Weir, B. S., W. G. Hill and L. R. Cardon. 2004. Allelic association patterns for a dense SNP map. Genet. Epidemiol. **27**: 442–50.

115 Weiss, K. M. and A. G. Clark. 2002. Linkage disequilibrium and the mapping of complex human traits. Trends Genet. **18**: 19–24.

116 Xiong, M. and S. W. Guo. 1997. Fine-scale genetic mapping based on linkage disequilibrium: theory and applications. Am. J. Hum. Genet. **60**: 1513–31.

117 Xiong, M., J. Zhao and E. Boerwinkle. 2002. Generalized T2 test for genome association studies. Am. J. Hum. Genet. **70**: 1257–68.

118 Xiong, M., J. Zhao and E. Boerwinkle. 2003. Haplotype block linkage disequilibrium mapping. Front. Biosci. **8**: a85–93.

119 Yu, K., C. C. Gu, M. Province, C. J. Xiong and D. C. Rao. 2004. Genetic association mapping under founder heterogeneity via weighted haplotype similarity analysis in candidate genes. Genet. Epidemiol. **27**: 182–91.

120 Zhang, K., P. Calabrese, M. Nordborg and F. Sun. 2002. Haplotype block structure and its applications to association studies: power and study designs. Am. J. Hum. Genet. **71**: 1386–94.

121 Zhang, K., F. Sun, M. S. Waterman and T. Chen. 2003. Haplotype block partition with limited resources and applications to human chromosome 21 haplotype data. Am. J. Hum. Genet. **73**: 63–73.

122 Zhao, H., S. Zhang, K. R. Merikangas, M. Trixler, D. B. Wildenauer, F. Sun and K. K. Kidd.

2000. Transmission/disequilibrium tests using multiple tightly linked markers. Am. J. Hum. Genet. **67**: 936–46.

123 ZHAO, J. Y., L. JIN AND M. XIONG. 2006. Nonlinear tests for genome wide association studies. Genetics **106**.

124 ZHENG, G., B. FREIDLIN, Z. LI AND J. L. GASTWIRTH. 2005. Genomic control for association studies under various genetic models. Biometrics **61**: 186–92.

Appendix A

Let $f(P)$ be a vector-valued nonlinear function of random vector P. Assume that the nonlinear function $f(P)$ satisfies regularity conditions which ensure that Theorem 3.3A in Serfling [87] holds. Then, $f(\hat{P})$ is asymptotically distributed as a multivariate normal distribution $N(f(P), \frac{1}{2n_G} C\Sigma C^{\mathrm{T}})$, where

$$c_{ii} = \frac{\partial f P_{H_i}}{\partial (P_{H_i})}\ , \quad c_{ij} = \frac{\partial f(P_{H_i})}{\partial P_{H_j}}\ , \quad C = (c_{ij})_{m\times m}\ ,$$

$$\Sigma = \mathrm{diag}(P_{H_1}, \ldots, P_{H_m}) - PP^{\mathrm{T}}\ .$$

Similarly, $f(\hat{P}^A)$ is asymptotically distributed as $N(f(P^A), \frac{1}{2n_A} B\Sigma^A B^{\mathrm{T}})$, where

$$b_{ii} = \frac{\partial f(P^A_{H_i})}{\partial P^A_{H_i}}\ , \quad b_{ij} = \frac{\partial f(P^A_{H_i})}{\partial P^A_{H_j}}\ , \quad B = (b_{ij})_{m\times m}\ ,$$

$$\Sigma^A = \mathrm{diag}(P^A_{H_1}, \ldots, P^A_{H_m}) - P^A(P^A)^{\mathrm{T}}\ .$$

Therefore, under the null hypothesis $H_0 : P^A = P$, which implies $f(P^A) = f(P)$, $f(\hat{P}^A) - f(\hat{P})$ is asymptotically distributed as $N(0, \Lambda)$, where:

$$\Lambda = \frac{1}{2n_A} B\Sigma^A B^{\mathrm{T}} + \frac{1}{2n_G} C\Sigma C^{\mathrm{T}}\ .$$

Let $Z = f(\hat{P}^A) - f(\hat{P})$ and $r = \mathrm{rank}(\Lambda)$. Then, under the null hypothesis, $T_{\mathrm{N}} = Z^{\mathrm{T}}\Lambda^{-}Z$ is asymptotically distributed as a central $\chi^2_{(r)}$ distribution [43]. The alternative hypothesis is $H_a : P^A \neq P$. Under the alternative hypothesis, T_{N} is asymptotically distributed as a noncentral $\chi^2_{(r)}$ distribution with the following noncentrality parameter:

$$\lambda_{\mathrm{N}}[f(P^A) - f(P)]^{\mathrm{T}}\Lambda^{-}[f(P^A) - f(P)] \tag{A.1}$$

By Taylor expansion, we have:

$$f(P^A) - f(P) \approx C(P^A - P) + \frac{1}{2}(P^A - P)^{\mathrm{T}} H(P^A - P) \tag{A.2}$$

where $H_1 = \operatorname{diag}(0,\ldots,f''(P_{H_l}),0\ldots0), l = 0,\ldots,m$

$$(P^A - P)^{\mathrm{T}} H (P^A - P) = \begin{bmatrix} (P^A - P)^{\mathrm{T}} H_1 (P^A - P) \\ \vdots \\ (P^A - P)^{\mathrm{T}} H_m (P^A - P) \end{bmatrix}$$

Equation (A2) can be rewritten as:

$$\begin{aligned} f(P^A) - f(P) &\approx C\left[(P^A - P) + \frac{1}{2}C^{-}(P^A - P)^{\mathrm{T}} H (P^A - P)\right] \\ &= C\left[(P^A - P) + \frac{1}{2}C^{-} H (P^A - P)(P^A - P)\right] . \end{aligned} \tag{A.3}$$

Let $S = C^{-}H(P^A - P)$, then:

$$f(P^A) - f(P) \approx C\left(I + \frac{1}{2}S\right)(P^A - P) . \tag{A.4}$$

Substituting $f(P^A) - f(P)$ in Eq. (A4) into Eq. (A1) yields:

$$\begin{aligned} \lambda_{\mathrm{N}} &= (P^A - P)^{\mathrm{T}}\left(I + \frac{1}{2}S\right)^{\mathrm{T}}(C^{\mathrm{T}}\Lambda^{-}C)\left(I + \frac{1}{2}S\right)(P^A - P) \qquad \text{(A.5)} \\ &= (P^A - P)^{\mathrm{T}}\left(I + \frac{1}{2}S\right)^{\mathrm{T}}\left[(C^{-}\Lambda(C^{\mathrm{T}})^{-}\right]^{-}\left(I + \frac{1}{2}S\right)(P^A - P) . \end{aligned}$$

Recall that:

$$P^A - P = e\delta_{\mathrm{HD}} \quad \text{and} \quad B \approx C + H(P^A - P) \tag{A.6}$$

where $\delta_{\mathrm{HD}} = [\delta_{H_1 D},\ldots,\delta_{H_m D}]^{\mathrm{T}}$.

Thus:

$$\begin{aligned} C^{-}\Lambda(C^{\mathrm{T}})^{-} &= C^{-}\left[\frac{1}{2n_A}B\Sigma^A B^{\mathrm{T}} + \frac{1}{2n_G}C\Sigma C^{\mathrm{T}}\right](C^{\mathrm{T}})^{-} \\ &= \frac{1}{2n_A}C^{-}B\Sigma^A(C^{-}B)^{\mathrm{T}} + \frac{1}{2n_G}\Sigma \\ &= \frac{1}{2n_A}(I + S)\Sigma^A(I + S) + \frac{1}{2n_G}\Sigma . \end{aligned} \tag{A.7}$$

Substituting Eqs. (A6) and (A7) into Eq. (A5), we obtain:

$$\lambda_{\mathrm{N}} \approx e^2\delta_{\mathrm{HD}}^{\mathrm{T}}\left(I + \frac{1}{2}S\right)^{\mathrm{T}}\left[\frac{1}{2n_A}(I + S)\Sigma^A(I + S) + \frac{1}{2n_G}\Sigma\right]^{-}\left(I + \frac{1}{2}S\right)\delta_{\mathrm{HD}} . \tag{A.8}$$

Next we study geometric interpretation of the matrix S. Let $\gamma(P) = Z_1, \dots, Z_m]^{\mathrm{T}}$, where $Z_i = f(P_{H_i})$. Define the following parameter equations:

$$P^A = P + t\Delta P\,.$$

As t varies, $\gamma(P^A) = \gamma(P + t\Delta P)$ defines a curve C in the space. The tangent vector of the curve C at the point P is given by:

$$\frac{\mathrm{d}\gamma}{\mathrm{d}t} = \frac{\partial\gamma}{\partial P^{\mathrm{T}}}\Delta P\,, \quad \text{where} \quad \frac{\partial\gamma}{\partial P^{\mathrm{T}}} = \begin{bmatrix} f'(P_{H_1}) & 0 & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & f'(P_{H_m}) \end{bmatrix}.$$

Taking Z_i $(i = 1, \dots, m)$ as a new coordinate system, we obtain the change rates of the tangent vector of the curve over new coordinates:

$$\frac{\partial\left(\frac{\mathrm{d}r}{\mathrm{d}t}\right)}{\partial Z^{\mathrm{T}}} = C^{-}H\Delta P = S\,, \quad \text{where} \quad H = \begin{bmatrix} H_1 \\ \vdots \\ H_m \end{bmatrix}$$

$$\text{and} \quad H_i = \mathrm{diag}(0, \dots, f''(H_i), \dots, 0)$$

The change rate of the tangent vector of the curve characterizes the strength of the nonlinearity of the nonlinear function [5]. The vector S has the following form:

$$S = \mathrm{diag}\left(\frac{f''(P_{H_1})}{f'(P_{H_1})}(P^A_{H_1} - P_{H_1}), \dots, \quad \frac{f''(P_{H_m})}{f'(P_{H_m})}(P^A_{H_m} - P_{H_m})\right)\,.$$

If the product terms of the haplotype frequencies are ignored, we obtain $C^{-}\Lambda(C^{\mathrm{T}})^{-} = \mathrm{diag}(\Lambda_1, \dots, \Lambda_m)$, where

$$\Lambda_i = \frac{1}{2n_A}[1 + \pi_i(P^A_{H_i} - P_{H_i})]^2 P^A_{H_i} + \frac{1}{2n_G}P_{H_i}\,, \quad \pi_i = \frac{f''(P_{H_i})}{f'(P_{H_i})}\,,$$

$$I + \frac{1}{2}S = \mathrm{diag}\left(1 + \frac{\pi_1}{2}(P^A_{H_1} - P_{H_1}), \dots, \ 1 + \frac{\pi_m}{2}(P^A_{H_m} - P_{H_m})\right)\,.$$

Then, Eq. (A8) can be simplified to:

$$\begin{aligned} \lambda_{\mathrm{N}} &\approx e^2 \sum_{i=1}^{m} \frac{\delta^2_{H_iD}\left[1 + \frac{\pi_i}{2}(P^A_{H_i} - P_{H_i})\right]}{\Lambda_i} \\ &= e^2 \sum_{i=1}^{m} \frac{\delta^2_{H_iD}\left[1 + \frac{e\pi_i}{2}\delta_{H_iD}\right]^2}{\frac{1}{2n_A}\left(1 + \frac{e\pi_i\delta_{H_iD}}{2}\right)^2 P^A_{H_i} + \frac{1}{2n_G}P_{H_i}}\,. \end{aligned}$$

For the standard χ^2 test statistic, we have $\pi_i = 0$. Thus, its noncentrality parameter is given by:

$$\lambda \approx e^2 \delta_{\mathrm{HD}}^{\mathrm{T}} \left[\frac{1}{2n_A} \Sigma^A + \frac{1}{2n_G} \Sigma \right]^{-} \delta_{\mathrm{HD}} \quad \text{and} \quad \lambda \approx e^2 \sum_{i=1}^{m} \frac{\delta_{H_i D}^2}{\frac{1}{2n_A} P_{H_i}^A + \frac{1}{2n_G} P_{H_i}} \, .$$

39 Pharmacogenetics/Pharmacogenomics

Xing Jian Lou, Russ B. Altman, and Teri E. Klein

1 Introduction

Pharmacogenetics is the study of how variations in the genes of an organism affect its response to drugs. Since its inception in the mid-20th century [32, 45, 64, 69], the major goal of pharmacogenetics research has been to maximize drug efficacy and minimize drug toxicity. In recent years, with the sequencing of the human genome and the development of high-throughput genotyping methods, the ability to correlate genetic variations to drug responses has dramatically increased. Therefore, pharmacogenetics has been moving from the study of the impact of single-gene variations on drug response to the study of the entire complement of pharmacologically relevant genes (*pharmacogenes*), and how these genes and their variations interact with each other and the environment to affect drug response. Although "pharmacogenetics" and "pharmacogenomics" are often used interchangeably, the term *pharmacogenomics* is usually associated with the collection of data, including DNA sequence variations and mRNA expression, using large-scale and high-throughput methods. Pharmacogenomics research yields complex data that are relevant to many genes and difficult to integrate. Traditional methods of manual data collection and management are no longer cost-effective with the development of high-throughput pharmacogenetics and pharmacogenomics experiments. In this chapter, we review the challenges that pharmacogenomics brings to biomedical informatics as well as biomedical informatics tools that can catalyze the research and development of pharmacogenetics and pharmacogenomics.

2 An Overview of Pharmacogenetics and Pharmacogenomics

The concept of pharmacogenetics originated from the clinical observation that plasma or urinary drug concentrations were variable and could be inherited. Through more than 50 years of research, pharmacogenetics has

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

provided knowledge on how the genetic variation of single genes affects the drug toxicity and efficacy for drug development and therapy. Despite this progress, many multidimensional relationships between genes, drugs, disease and environment remain unclear. Pharmacogenomics studies using genomic approaches are expected to accelerate the process of understanding these multiple interactions.

2.1 Background of Pharmacogenetics and Pharmacogenomics

The classic example of pharmacogenetics is as follows. A 4-year-old boy has acute lymphoblastic leukemia. His treatment protocol includes daily doses of oral 6-mercaptopurine (6MP). However, this treatment leads to excess toxicity, including severe life-threatening bone marrow suppression. The physician performs a genetic test for thiopurine S-methyltransferase (TPMT) and learns that the patient has two mutated copies of this gene that cannot metabolize 6MP into its usual inactive metabolite. Instead, the 6MP is metabolized by a different pathway that leads to toxic compounds, called 6-thioguanines. Based on this information, his physician greatly reduces his dose of 6MP and monitors the resulting response to the drug. The boy subsequently has an uneventful several-year maintenance period and achieves complete remission [21].

Before the genetic variations of TPMT, an enzyme that metabolizes 6MP, were discovered, physicians had long wondered why approximately one in 300 Caucasian patients had serious, sometimes lethal, myelosuppression during the treatment with regular dosages of 6MP [67, 68]. Now we know that this adverse drug response results from a lack of TPMT as a result of a genetic alteration. When there is insufficient TPMT, 6MP is metabolized to 6-thioguanine nucleotide (6TGN), a toxic 6MP metabolite, instead of its normal metabolite, 6-methly-MP [36–38, 68]. Severe side-effects and adverse reactions then occur under the "standard-of-care" treatment. There is also evidence that an increased activity of TPMT is associated with a decreased efficacy of 6MP [37].

Most initial pharmacogenetic discoveries involved a small number of polymorphisms (variations in genes, see also Chapter 37) in a single gene associated with a very dramatic change in the drug response. These simple, but striking, examples provided the foundation for our present understanding that inheritance can play an important role in individual variations in drug response by influencing efficacy, toxicity or both. Drug-metabolizing enzymes are divided into two large subgroups: phase I (functionalization) and phase II (conjugation) enzymes. Phase I reactions consist of oxidation, reduction and hydrolysis. These reactions usually lead to metabolites that are more polar than the parent compounds. In a phase II reaction, an endogenous hydrophilic moiety is attached to a target molecule, producing a metabolite that is more water soluble than the parent compound. Glucuronidation, sulfation, acetylation and conjugation to glutathione and amino acids are the major conjugation reactions. The most common variations associated with drug responses reported occur in genes coding enzymes of both phase I and phase II metabolism [66]. Polymorphisms in cytochrome P450 genes (such as CYP2D6, one of the phase I genes) are the most widely studied and best understood, because these polymorphisms affect the metabolism of a wide range of commonly used drugs [12, 53]. Polymorphisms in genes from phase II metabolism, such as TPMT and *N*-acetyltransferases (NATs), have also been shown to affect drug response. Transporter gene polymorphisms are also found to be associated with drug efficacy and toxicity [60]. Recently, there has been a focus on polymorphisms in genes that are the targets of drug therapy (and are not involved in metabolism). For example, mutations in the epithelial growth factor receptor (EGFR) gene have been shown to enhance tumor response to the EGFR tyrosine kinase inhibitor gefitinib (Iressa; AstraZeneca) [41, 46].

Like many other genetic phenomena, the response to many drugs is determined by the interaction of multiple genes at different loci [31, 47]. In addition, environmental factors including diet, age and lifestyle also play important roles in a person's response to drug treatment. Environmental factors need to be studied in the context of an individual's genetic make-up in order to understand their effects [24, 29]. Pharmacogenetics has thus expanded to pharmacogenomics and includes multidisciplinary research efforts, including genetics, molecular biology, cell biology, physiology, pharmacology, biochemistry, toxicology, clinical pharmacology, clinical pharmacy, epidemiology, pharmaceutical sciences and bioinformatics.

2.2 Influence of Pharmacogenetics and Pharmacogenomics on Drug Development and Therapy

Most pharmaceutical companies avoid developing drugs that are metabolized primarily by polymorphic enzymes such as CYP2D6 because of interindividual variation in response. However, this precaution is not always effective, especially when variations exist in the drug targets. Therefore, an important aspect of drug development is to develop pharmacogenetics tests to define the patient population that will benefit most from the drug treatment with minimal adverse reactions. Using pharmacogenetics tests to select the right patient group may significantly decrease the cost of drug development. Clinical trials focused on the right population (responders without adverse reaction) may decrease the trial size and increase the success rate. Although there is still debate about the cost-effectiveness of genetic screening before drug trials, there is evidence that genetic variations can be used to enrich for normal and hyper-responders to decrease the size of clinical trials [54]. A population-based study of genetic variations associated with increased risk of disease conducted by deCODE Genetics has also led to the discovery of a disease-specific pathway and shortened the time of target discovery through the phase II clinical trial to little more than 1 year [23,27,28].

On the therapy side, few currently available medicines actually treat *all* patients effectively. Some medicines are licensed on the basis of only 30% efficacy in clinical trials [57]. Physicians have therefore accepted in practice that many medicines are prescribed by trial and error. Patients first take one medicine and, if it is shown to be ineffective, are then prescribed another. Unfortunately, the adverse risks that are associated with medicines can be lethal and can be additive. Adverse reactions represent the fourth to the sixth leading cause of death in the US [7,51]. Therefore, it would be useful to determine, before prescribing a drug, whether a patient is likely to respond, and if that individual is at risk of adverse reactions. In the US there are only three pharmacogenetic tests that have been approved by the US Food and Drug Administration (FDA) to be used by physicians to personalize treatment decisions for maximizing efficacy and avoiding adverse reactions. AmpliChip of Roche is used to individualize the dosage of antidepressants, antipsychotics, β-blockers, and some chemotherapy drugs by detecting gene variations for the CYP2D6 and CYP2C19 genes. The TRUGENE HIV-1 identifies variations in the viral gene that make the virus resistant to some antiretroviral drugs. Most recently, the Invader UGT1A1 test of Third Wave Technologies was approved for identifying patients with specific mutations in the UGT1A1 gene. These patients may be at increased risk of adverse reaction to chemotherapy drugs such as irinotecan. Although the number of pharmacogenetic tests is currently limited, a lot of effort and resources have been put into this field by private, public and regulatory sectors such as the Roche decode alliance (http://www.decode.com) and the ScanBalt Clinical Research Network (http://www.scanbalt.org/sw229.asp). That two out of three pharmacogenetic tests were approved by the FDA in 2005 indicates the rapid progress and growth in this area.

3 Biomedical Informatics Resources Relevant to Pharmacogenomics

The explosion of biological information in the last decade has triggered the establishment of many successful biological databases in areas such as sequences [5,48], structures [15] and functions [4,33]. These databases provide basic data of relevance to pharmacogenomic. The Reference Sequence (RefSeq) Project at the National Center for Biotechnology Information (NCBI) (http://www.ncbi.nlm.nih.gov/RefSeq) provides a comprehensive and nonredundant set of sequences, including genomic DNA, transcript (RNA) and protein products for major research organisms. It also serves as a standard for sequence annotation such as exon, intron, alternative splicing isoforms, and 3′- and 5′-untranslated regions of genes [48]. The human genome browsers offered by University of California at Santa Cruz (UCSC) (http://www.genome.ucsc.edu), Ensembl (http://www.ensembl.org) and NCBI (http://www.ncbi.nlm.nih.gov/mapview/map_search.cgi) offer comprehensive information about gene location and genomic annotation. The Single Nucleotide Polymorphism database (dbSNP) at NCBI (http://www.ncbi.nlm.nih.gov/projects/SNP/index.html) and International HapMap Project [1,2] provide excellent sources for the locations and types of genetic variations of reported SNPs, including those submitted by The SNP Consortium [56,63] – an industrial group that is performing large-scale SNP screening.

The success of pharmacogenomics depends on the description of functional features (phenotypes) associated with gene products. These phenotypes range in detail from the molecular [19] to the individual [36] and population levels [41,43], including gene expression profiles, enzyme kinetic data, blood pressure and drug responses in a particular population of patients. Some progress has been achieved in the representation of these phenotypes. Efforts have begun in providing standards of metabolism and signaling pathway presentations [10,30,33] (BioPax: http://www.biopax.org/index.html, see also Chapters 20 and 22). Furthermore, existing standards for coding diagnoses (International Classification of Diseases: http://www3.who.int/icd/vol1htm2003/fr-icd.htm; MeSH: http://www.nlm.nih.gov/mesh), pathology (Systematized Nomenclature of Medicine – SNOMED: http://www.snomed.org) and procedures (Current Procedural Terminology: http://www.ama-assn.org/ama/pub/category/3113.html) in clinical medicine offer a good starting point for pharmacogenomics research. However, these clinical data representations do not provide the precision required for high-quality data retrieval. Also there is a fair amount of agreement within the pharmacology community on how to represent pharmacokinetic profiles. Programs such as ADAPT II [9] and NONMEM (http://c255.ucsf.edu/nonmem0.html) allow first- and second-order kinetics and associated parameters to be computed. A standard set of parameters, including the dissociation constant for binding of inhibitor to enzyme (K_i), the concentration of substrate that produces half-maximal velocity (K_m) and the maximal velocity of a reaction (V_max), have fairly consistent definitions and thus provide a good initial opportunity for modeling of the data (see also Chapters 20 and 22).

A set of controlled terms is called a standard vocabulary or controlled vocabularies. The Human Genome Organization (HUGO) Nomenclature Committee (HGNC) has created a reference set of symbols for human genes [65]. These symbols are used by almost all databases that involve genes or gene products (e.g. protein). Therefore, this nomenclature serves as a standard vocabulary for pharmacogenomics regarding genes. Controlled vocabularies are also very useful for indexing purposes. For example, whereas an abstract in Medline is free text, the list of Medical Subject Heading (MeSH) keywords from the Medline database (http://www.ncbi.nlm.gov/PubMed/) represents a controlled vocabulary. The advantage of a controlled vocabulary is that computer programs can be written to detect certain phrases and can be programmed to process data based on the occurrence of these phrases.

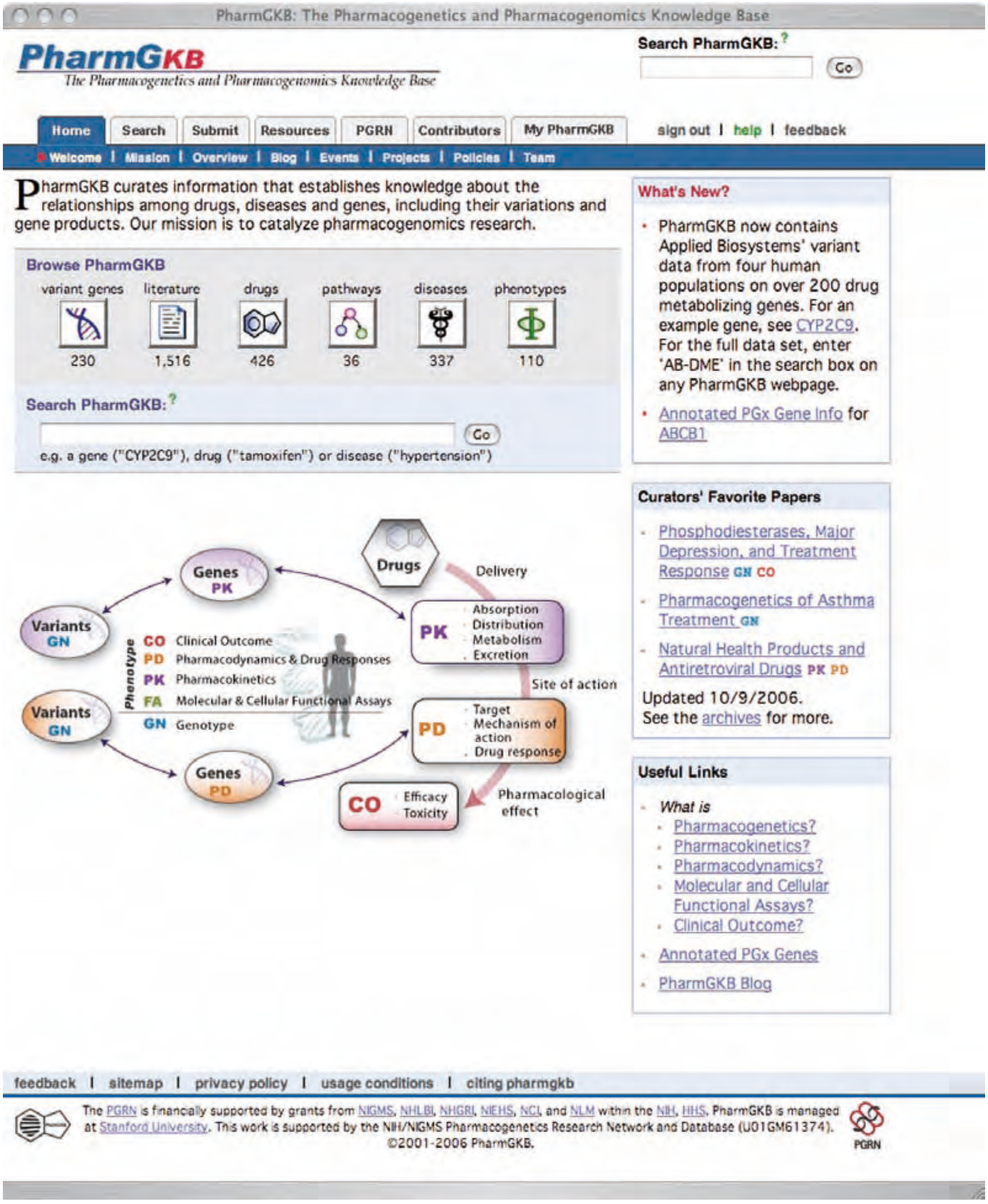
There have been a number of proposed standards for exchange sequence data, including standards that support GenBank submission such as BIOML (http://bioinformatics.genomicsolutions.com/BioML.html), Bioperl (http:// bio. perl.org) and GCG (http://www.accelrys.com/products/gcg_wisconsin_package). There has also recently been much effort in creating a standard for exchange of genetic polymorphism information, such as the PharmGKB eXtensible Markup Language (XML) schema (http://www.pharmgkb.org/schema/index.html) and the Polymorphism Markup Language (PML) [61]. The Microarray Gene Expression Data (MGED) society also has developed Microarray and Gene Expression Markup Language (MAGE-ML) – a data exchange format for microarray expression experiments [58]. MGED is an international organization of biologists, computer scientists and data analysts that aims to facilitate the sharing of microarray data generated by functional genomics and proteomics experiments. MAGE-ML specifies a set of Minimum Information About a Microarray Experiment (MIAME), including experimental conditions, the quality control parameters, the list of genes being assayed and the actual expression measurements (and background measurements) recorded, and is implemented using XML (see also Chapter 23). This format has been widely accepted by the microarray and gene expression community and can be used for pharmacogenomics studies as well.

While data exchange format and controlled vocabulary are well developed for microarray data and most of the sequence-related genomic data, many pharmacogenomic, especially phenotype, data related to function still lack such standards. The mouse genome initiative has created an ontology that provides a general framework for communication on mouse phenotypes (http://www.informatics.jax.org/searches/MP_form.shtml). With the objective of capturing information about phenotypes in any organism, in 2002, Ashburner also proposed the Phenotype And Trait Ontology (PATO: http://obo.sourceforge.net). Ontologies based on the PATO proposal are under development [20]

4 Building the PharmGKB

The Pharmacogenetics and Pharmacogenomics Knowledge Base (PharmGKB, http://www.pharmgkb.org) is a National Institutes of Health (NIH)-funded effort to build a centralized repository for genetic and clinical information on all individuals participating in pharmacogenomics studies as well as for molecular, cellular and pharmacological information on model systems that are used for pharmacogenomics research. The PharmGKB is currently being developed at Stanford University. It is web based, and supports the storage, integration and dissemination of data and knowledge about pharmacogenetics and pharmacogenomics. The home page of PharmGKB is shown in Figure 1. In brief, the PharmGKB classifies primary phenotype and genotype data sets submitted by the pharmacogenetics and pharmacogenomics community, described below, into five categories: (i) clinical outcome, (ii) pharmacodynamics and drug responses, (iii) pharmacokinetics, (iv) molecular and cellular functional assays, and (v) variation in genetic sequence. Users browse data by associated genes, drugs, diseases and pathways. These pathways show how drugs are metabolized (pharmacokinetics) and how drugs act (pharmacodynamics). In addition to primary genotype and phenotype data sets, PharmGKB collects information about gene–drug–disease interactions of relevance to pharmacogenetics from the literature. Using controlled vocabularies and standard exchange formats, PharmGKB integrates this data with relevant information from other databases, and provides browsing, searching and analytical functions to help scientists discover connections between genetic variations and alterations in drug responses and related phenotypes.

Different approaches are used at facilities within the pharmacogenetics and pharmacogenomics community to uncover the mechanisms underlying the inter-individual differences in the response to drug treatments. For example, some groups specialize on the study of genetic variations affecting the treatment of a particular disease (e.g. http://www.pharmgkb.org/network/members/parc.jsp), some groups focus on studying the influence of multiple genetic effects on a single drug (e.g. http://www.pharmgkb.org/network/members/cobra.jps), and others on the study of variations of a group of genes and their resulting effect on drug treatment (e.g. http://www.pharmgkb.org/network/members/pmt.jsp). Studied diseases include, but are not limited to, arrhythmia, hypertension, atherosclerosis, asthma, cancer and depression. Examples of investigated drugs include tamoxifen, statin, ACE inhibitors and irinotecan. Numerous functional polymorphisms on genes responsible for drug transport, metabolism and genes targeted by drugs are under investigation in pharmacogenetics and pharmacogenomics study centers.



PharmGKB curates information that establishes knowledge about the relationships among drugs, diseases and genes, including their variations and gene products. Our mission is to catalyze pharmacogenomics research.

e.g. a gene ("CYP2C9"), drug ("tamoxifen") or disease ("hypertension")

What's New?

- PharmGKB now contains Applied Biosystems' variant data from four human populations on over 200 drug metabolizing genes. For an example gene, see CYP2C9. For the full data set, enter 'AB-DME' in the search box on any PharmGKB webpage.
- Annotated PGx Gene Info for ABCB1

Curators' Favorite Papers

- Phosphodiesterases, Major Depression, and Treatment Response GN CO
- Pharmacogenetics of Asthma Treatment GN
- Natural Health Products and Antiretroviral Drugs PK PD

Updated 10/9/2006.
See the archives for more.

Useful Links

- What is
  - Pharmacogenetics?
  - Pharmacokinetics?
  - Pharmacodynamics?
  - Molecular and Cellular Functional Assays?
  - Clinical Outcome?
- Annotated PGx Genes
- PharmGKB Blog

The PGRN is financially supported by grants from NIGMS, NHLBI, NHGRI, NIEHS, NCI, and NLM within the NIH, HHS. PharmGKB is managed at Stanford University. This work is supported by the NIH/NIGMS Pharmacogenetics Research Network and Database (U01GM61374).
©2001-2006 PharmGKB.

Figure 1 The PharmGKB home page. Contents of the Knowledge Base are highlighted in the gray shaded box.

In addition to the PharmGKB, there are other tools available for specialized topics in pharmacogenomics. For example, the Pharmacogenomics of

Arrhythmia Therapy group at Vanderbilt University has developed a data reduction algorithm [Multifactor Dimensionality Reduction (MDR)] for the detection of gene–gene, gene–drug or gene–environment interactions in relatively small sample sizes [22,52]. The number of analytical and computational approaches continues to grow with the challenges facing the pharmacogenomics community.

PharmGKB uses the SOAP (simple object access protocol) web-service standard method (http://www.pharmgkb.org/home/projects/webservices/index.jsp) to allow users to access these analytical and computational tools (see also Chapter 43 for bioinformatics web technology). For example, PharmGKB has demonstrated a service on the gene variant page that triggers code to gather all variants for that gene, format them as required by the haplotype inference algorithm PHASE and ship the data to the Channing Laboratory in Boston for computation. Users therefore are able to access data through the central repository of PharmGKB, but use tools developed by third parties to analyze such data through PharmGKB's web service.

General information in the PharmGKB is available to the public without restriction. However, genotypic and phenotypic information associated with individual human subjects is only available to users that have been granted access, i.e. "registered". Patient privacy policies are discussed in more detail in Section 4.3.3.

4.1 Establishing a Repository of Pharmacogenetics and Pharmacogenomics Information

Pharmacogenetics and pharmacogenomics research usually involves the analysis of multidimensional data obtained through different approaches and using different technologies. Integration is very important at many levels. For example, data on functional changes are meaningless if they are not associated with a particular genotype. The establishment of a central data repository and maintaining the relationships among these data provides a foundation for such integration.

4.1.1 The Data Model

A good data model is critical to the establishment of a successful pharmacogenomics knowledge base, because of the great diversity of information formats and the rapid evolution of the field. A data model contains important classes of objects in the domain of interest, the key attributes for these objects and the logical relationships among these objects.

The high-level objects in a pharmacogenomics data model are shown in Figure 2. Functional considerations for a database include the ability to accept data submissions from the user community, provide a preview of submitted

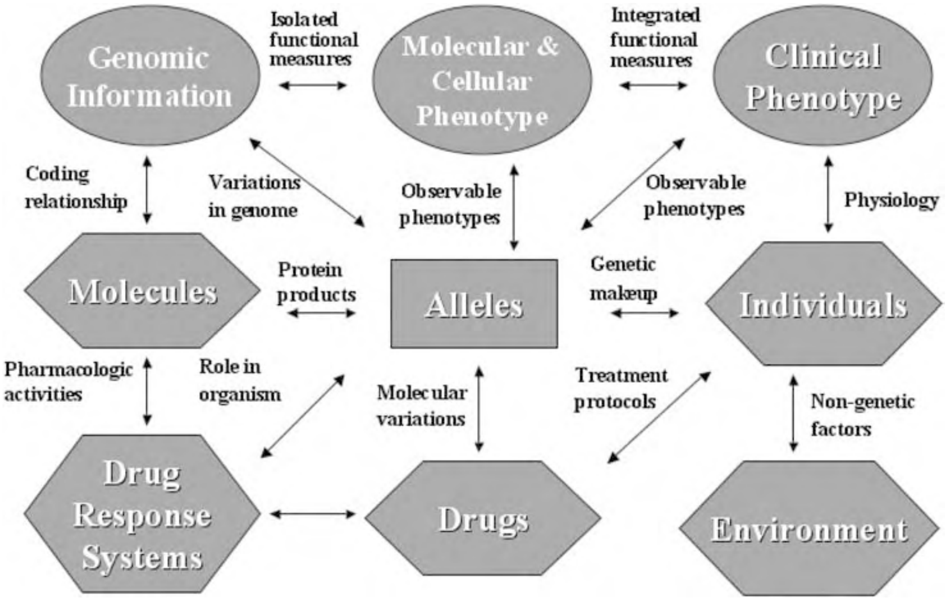


Figure 2 PharmGKB data model. Each gray box represents a class of objects that has a large tree of subclasses of objects underneath it for more specificity. The lines between the objects summarize the types of links that occur between instances of each type of objects.

data to the submitters for inspection and approval, provide access to all approved submissions to all registered users, provide a search capability to find major objects from any text field associated with the object, and track changes to the data and be able to retrieve a date dependent view of the data. More information about one sample pharmacogenomics relational data model can be viewed at http://www.pharmgkb.org/resources/references.jsp.

Relational database management systems are the most dependable and widely used architectures for building large databases. They provide built-in support for dynamic and hierarchical data modeling, including adding objects, associating particular attributes with each object and interlinking objects with named relationships. For example, PharmGKB is built using a relational database architecture upon which a middle layer (written in Java) contains all logic and a final presentation layer presents data to the user. The technical infrastructure of PharmGKB is detailed at http://www.pharmgkb.org/resources/references/architecture.jsp.

4.1.2 Primary Data

The main source of the pharmacogenomic primary data is from pharmacogenetics investigators. Information regarding the research interest and affiliations of all the groups to submitting to PharmGKB is available at http://www.pharmgkb.org/views/loadContributors.action.

Fundamentally, there are only two types of data generated from pharmacogenetic and pharmacogenomic research – genotype data and phenotype data. Genotype is the internally coded, heritable information carried by the organism. Variation in genotypes represents differences in sequence within a species, such as SNPs (a very common gene variation that only involves the change of a single nucleotide) the locations or the number of repeats, deletions, or critical splice sites (see also Chapters 36 and 37). Phenotypes are the observable properties of an organism produced by the interaction of the genotype with the environment. For pharmacogenetics, the "environment" is often defined via the exposure to a drug, although it may include other variables, e.g. describing smoking habits or alcohol consumption. However, because an individual's drug response is usually a consequence of interactions among many genes or proteins (encoded by genes), including transporters, enzymes and receptors in different cells, tissues and organs, "the observable properties of an organism" here can vary from the change in the expression of a transporter gene, across the increase or decrease of an enzyme's activity or the binding affinity of a receptor to how the individual feels, e.g. headache, diarrhea, etc. Different assays are designed to measure these different changes, including how these changes interact with each other and trigger a new change. Therefore, it is fair to say that there are as many phenotypes as assays that have been invented to measure them.

Individual primary phenotype files in PharmGKB are carefully curated with respect to clear documentation, scientific clarity, and privacy considerations. Approved files will be then labeled with:

- List of genes, drugs, diseases relevant to the file (using controlled vocabularies).
- List of index terms from the phenotype ontology.
- List of MeSH terms for additional indexing.

The approval and labeling procedure is an interactive process between curators and submitters. It is usually time consuming, but it is one of the necessary procedures to ensure the high quality of the data in the knowledge base.

4.1.3 Data from Literature

Thousands of papers related to pharmacogenetics and pharmacogenomics research have been published in the past decades. Numerous scientific discoveries and observations are recorded in these papers. Most of them are written in the text of natural language and stored in a database like Medline with a less-structured form to accommodate the heterogeneity of information.

Whereas natural language text has much power for expressiveness, it is difficult for computers to parse such language. For example, ideally a user would like to be able to compare results among some of the relevant papers then raise new hypotheses for further testing or to validate a discovery. Therefore, it is useful to retrieve information from these publications and store it in a database for facile comparison and integration. However, reliably retrieving information from these papers is a considerable challenge. Three approaches have been taken to solve this problem. First, one can use an online tool for the scientific community to submit published literature providing evidence for relationships between genes, drugs and diseases. We have developed a web interface that accepts HGNC gene names, MeSH disease names and VA-NDF, Apelon and USP DI® drug names as a summary of published literature finding. These annotations are associated with existing evidence (a PubMed ID or an URL) and a confidence measure for the evidence (select from low to high or unknown). Submitters are encouraged to categorize the submitted literature into one of the knowledge categories discussed in Section 4.2.1.

The second way to capture information in the literature is manual curation. Database curators routinely review the literature to find gene–drug–disease associations that should be captured. Curators survey the literature using PubMed, clinicaltrials.gov, the websites of the National Cancer Institute (NCI), National Heart Lung and Blood Institute (NHLBI) and Google.

While human (especially expert) knowledge is very important in collecting information in the pharmacogenetics and pharmacogenomics field, the process is very time consuming and therefore the amount of information collected through this method is relatively small. A third approach to acquire information from published literature is based on automatic computer algorithms to extract information automatically from text. We have developed a statistical and computational approach to identify all the articles in Medline citations that are associated with pharmacogenetics and pharmacogenomics research. This method can also be used to extract data pertaining to particular gene–drug relationships with 92% precision [55]. One can search for drugs that are associated with a gene of interest or *vice versa*. This tool is available at http://pharmdemo.stanford.edu/pharmdb/main.spy. For more details about tools for analysis of pharmacogenetics text, users can also visit http://bionlp.stanford.edu/genedrug. For automatic analysis of scientific texts, see also Chapter 33.

4.1.4 **Linking to other Data Resources**

Many biological databases provide data that is complementary to the basic genotype–phenotype data sets collected for pharmacogenomics studies. The key to making appropriate links between database resources is to have common terminologies for referring to basic concepts such as genes and drugs.

For example, using shared vocabularies such as the HGNC gene symbols, one can link via URL directly to the following resources:

- UCSC Genome Browser [34]
- Entrez Gene [42]
- GenBank [5]
- Swiss-Prot [4]
- PubMed [17]
- OMIM [25]
- MedlinePlus (http://medlineplus.gov)
- Gene Ontology (GO; http://www.ebi.ac.uk/GOA)
- SOURCE [16]
- PromoLign [74]
- RefSeq [48]
- Ensembl [6]
- GENATLAS [18]
- GeneCards [50]
- MutDB [14]
- PharmGKB

4.2 Turning Data into Knowledge

It is critical for a public repository of pharmacogenetics and pharmacogenomics data to present core data sets upon which researchers in the field can apply new analyses, generate new knowledge and build new hypotheses. In order to provide such data sets, data need to be organized and classified first so that one can browse and search for individual data set easily. Relationships among data sets should be maintained and/or enabled. These relationships should also be presented in biochemical pathways to best summarize connected pharmacogenetic and pharmacogenomic knowledge for generating new hypotheses and making new discoveries.

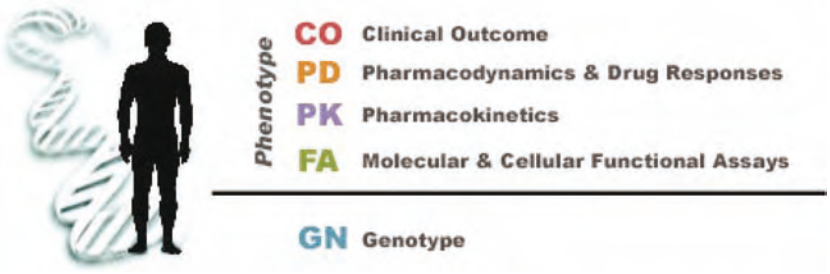

Phenotype
CO Clinical Outcome
PD Pharmacodynamics & Drug Responses
PK Pharmacokinetics
FA Molecular & Cellular Functional Assays
GN Genotype

Figure 3 Categories of pharmacogenetics knowledge used by PharmGKB.

4.2.1 **Categorizing Data**

As discussed in Section 4.1.2, phenotype data obtained in pharmacogenetics/pharmacogenomics research are extremely diverse. We have previously categorized pharmacogenetics and pharmacogenomics information based on two principles [3]. (i) Pharmacogenetics and pharmacogenomics relate variations in genes to variations in certain phenotypes associated with drugs. Therefore, gene variations, phenotype variations and drugs should be used as major labels for most pharmacogenetic and pharmacogenomic data sets. (ii) Phenotypic data can be further classified based on the level (i.e. molecular or whole organism) at which they are collected. The indexing schema containing the five classified variation information types is shown in Figure 3. These five categories are not perfectly separable, but they provide a framework for categorizing pharmacogenetic and pharmacogenomic data and literature. If data sets touch upon more than one area, they are multiply classified. These five categories have been used for indexing the published literature, labeling phenotype and genotype data sets, and designing the user interface for different types of users. Specific definitions of the five categories of data sets can be retrieved at PharmGKB through preprepared "simple queries" (http://www.pharmgkb.org/search/query/index.jsp), as detailed in the following subsections

4.2.1.1 **Genotype** The most basic information about pharmacogenetics and pharmacogenomics is the observation of variations in individual genes, including the type of the variation (i.e. SNP, insertion, deletion, etc.), the location of the variation on chromosomes (using, for example, the UCSC's Golden Path position) and the frequency of the variation in the populations of interest. Information related to the potential importance of the genetic variation or how to analyze the variants is also part of the genotype data. This includes subject information such as gender and ethnicity, features (such as exons and promoters) of the region under investigation, alternative splicing, and protocols for the assay by which the variant was analyzed. For example, genotype file PS204853 in PharmGKB contains the measurement of four variants in *VKORC1* genes that are critical to the therapeutic dose of warfarin, and the measurement was performed in 340 subjects with different age, sex and ethnicity background.

4.2.1.2 **Clinical Outcome**

The main driving force of pharmacogenetic and pharmacogenomic research is to make an impact on clinical medicine through maximizing drug efficacy and minimizing drug toxicity. Therefore, the discovery of genetic variations (genotype) that cause measurable differences in clinical outcomes that are of concern to patients and their physicians such as how tolerable is the treatment (drug side-effects, adverse reactions) or how soon the patient improves (disease symptoms, laboratory test results) in response to the drug treatment are particularly meaningful. These clinical outcomes are obviously different from the measurements of drug response in a research setting. Sometimes they might not be detailed enough to illustrate the pharmacodynamic effect of a drug, but they can certainly alter medical practice or policy. An example of clinical outcomes can be viewed in file PS204373 at PharmGKB. The rejection of a heart transplant in pediatric patients can be potentially managed by using specific genotype information of the *MDR1* and *CYP3A5* genes because such genotype information is important for maintaining the patient's blood concentration during immunosuppressive therapy with the drug tacrolimus.

4.2.1.3 **Pharmacodynamics and Drug Responses**

Datasets of measurements at the whole-organism level are classified into pharmacodynamics data and drug responses, although usually pharmacodynamics data can include all outcomes associated with how drugs act. For example, changes in an individual's intestinal CYP3A4 protein levels after 2 days of receiving oral rifampin are associated with variations in the pregnane X receptor (*PXR*) gene (phenotype file PS200308 in PharmGKB).

4.2.1.4 **Pharmacokinetics**

The measurements of the absorption, distribution, metabolism or elimination (ADME) of a drug in association with genotyping belong to the pharmacokinetics category. Classic pharmacogenetic studies relied upon observing changes in drug metabolism in the context of genetic variants. Up to now, most of the important genetic variants associated with drug responses are in genes coding enzymes involved in drug metabolism and transporters in charge of drug transportation. Many data sets have demonstrated that genetic polymorphisms lead to variation in the pharmacokinetics of particular drugs. For example, genetic variation in *VKORC1* is responsible for phenotypic variation in the steady-state concentration of warfarin in plasma (phenotype file PS402853 in PharmGKB).

4.2.1.5 Molecular and Cellular Functional Assays

Many discoveries of pharmacogenetics and pharmacogenomics start from measurements establishing an association between genetic variation and a drug induced or inhibited reaction at the molecular and cellular level. For example, the phenotype file PS200308 in PharmGKB, the example used in Section 4.2.1.3, reports data about the binding ability of the variants PXR*1, PXR*2, PXR*3 and PXR*4 of the PXR protein to CYP3A4 in a transient plasmid expression system. Data obtained *in vitro* with artificial constructs belong to the category of molecular and cellular functional assays.

4.2.2 Establishing Genotype–Phenotype Correlation

Correlating genotype findings to phenotype results is a key step to understanding the impact of genetic variations on drug response. Therefore, providing reliable information that links genotypes to phenotypes is an important goal.

Pharmacogenomics data contains many levels of complexity, including information on individual subjects on whom pharmacogenetics and pharmacogenomics assays have been performed. Assays performed on individual subjects include all of the genotyping efforts (using DNA samples) and most of the phenotypic studies (all of the *in vivo* and some of the *in vitro* studies on derived cell lines). These subjects might participate in multiple studies in different institutes. Thus, it is possible that genotype data can be submitted by one institute, pharmacokinetic phenotype data can be submitted by a different institute, and pharmcodynamic and drug responses phenotype data can be submitted by yet another institute. While ensuring that the privacy of these individuals is protected, it is important to keep track of all subjects in association with the submitted genotype or phenotype results. This information provides a way to link genotype data sets to data sets with different types of phenotypes. Users can view the genetic variations in these individuals and the phenotypic consequences of those variations. This provides one of the most reliable ways to study the potential correlations between genotypes and phenotypes.

Statistical and computational tools are needed to assist users in finding correlation between genotype and phenotype. For instance, a simple genotype–phenotype association can look for "single SNP, single phenotype measurement" relationships. An implementation could compute correlations between genotypic data and phenotypic data for biallelic SNP sites. These correlations can be computed on individual study populations, as well as on all available subjects pooled together. For more detailed analyses, users must be able to download data sets and perform more sophisticated correlation analysis as needed.

4.2.3 Using Pathways to Summarize Current Pharmacogenetics and Pharmacogenomics Knowledge

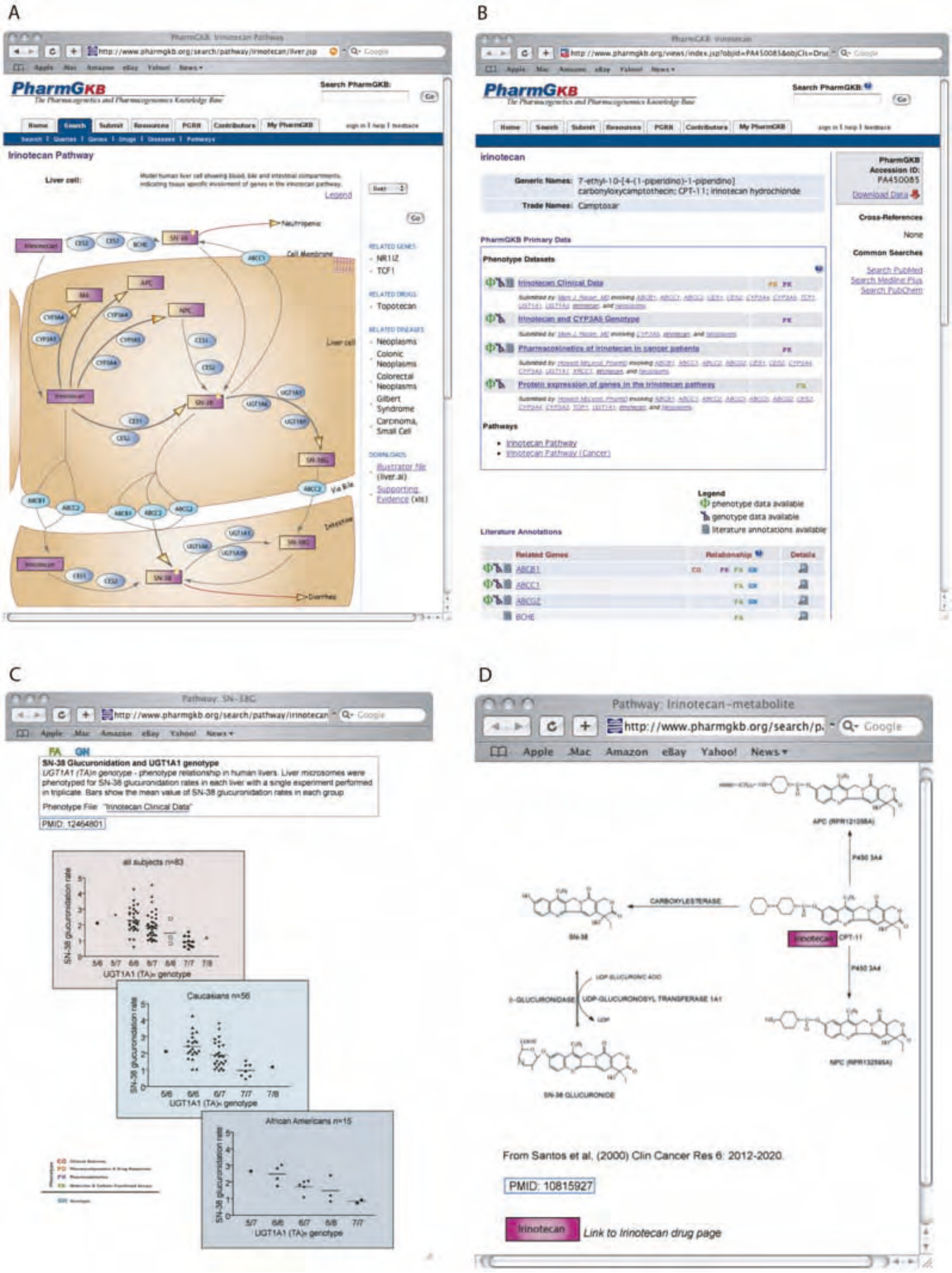
Pathways provide an excellent way to summarize connected pharmacogenetic and pharmacogenomic knowledge. In particular, it is critical to manage pathways related to drug metabolism or action. Users require both graphic and computational representations of how drugs are transported and metabolized (pharmacokinetics) and how drugs act (pharmacodynamics). In PharmGKB, pathway knowledge is linked with the underlying data in PharmGKB. Drugs, metabolites and transporter genes are presented using standard icons. Pathway representations are interactive – every gene and drug in a pathway diagram is linked to a related page which reports more detailed information regarding the gene and drug. The arrows in a pathway diagram indicate that a gene is involved in some process and are linked with the appropriate literature annotation or primary data that establishes or confirms this relationship.

In addition to the pathway diagram, a text summary is provided to describe the content on the graph and, more importantly, the drug-related significance of the pathway, the limitations of that particular graphic representation and any additional information not able to be included in the pathway diagram. Related pathways, drugs and diseases (that are not directly involved in this pathway but are related) are also listed next to the pathway. An example of such pathways is shown in Figure 4.

As pathways rely on expert knowledge, the creation of drug-related pathways requires the collection of relevant information and assessment by experts from the pharmacogenetic research community. PharmGKB has developed a protocol for curation of pathways by pharmacogenetics experts. A PharmGKB curator is assigned to the process. On a well-defined timeline, a draft pathway is produced, there are one or more conference calls to discuss it, and, when there is general agreement, the final pathway is rendered by a graphical interface specialist and posted on the website. It is also stored in a database using a simple relational representation, compatible with the emerging pathway standards (such as BioPAX). The most time-consuming part of this process is the documentation of standard names for drugs and genes, and the collection of references to the supporting data for the links to be portrayed. The assembly of a pathway typically takes 2–3 months.

4.3 Providing Easy Access of Knowledge for the Research Community

Easy access of knowledge and data can be obtained via the web-based querying and browsing systems. Most data submission can also be web-based.



PharmGKB: Irinotecan Pathway

http://www.pharmgkb.org/search/pathway/irinotecan/liver.jsp

PharmGKB
The Pharmacogenetics and Pharmacogenomics Knowledge Base

Search PharmGKB:

sign in | help | feedback

Search | Queries | Genes | Drugs | Diseases | Pathways

Irinotecan Pathway

Liver cell:

Model human liver cell showing blood, bile and intestinal compartments, indicating tissue specific involvement of genes in the irinotecan pathway.

Legend

liver

RELATED GENES
- NR1I2
- TCF1

RELATED DRUGS
- Topotecan

RELATED DISEASES
- Neoplasms
- Colonic Neoplasms
- Colorectal Neoplasms
- Gilbert Syndrome
- Carcinoma, Small Cell

DOWNLOADS
- Illustrator file (liver.ai)
- Supporting Evidence (xls)

PharmGKB: irinotecan

http://www.pharmgkb.org/views/index.jsp?objId=PA450085&objCls=Drug

irinotecan

Generic Names:	7-ethyl-10-[4-(1-piperidino)-1-piperidino] carbonyloxycamptothecin; CPT-11; irinotecan hydrochloride
Trade Names:	Camptosar

PharmGKB Accession ID: PA450085

Download Data

Cross-References

None

Common Searches

Search PubMed
Search Medline Plus
Search PubChem

PharmGKB Primary Data

Phenotype Datasets

Irinotecan Clinical Data — PD PK

Submitted by: Mark J. Ratain, MD involving ABCB1, ABCC1, ABCG2, CES1, CES2, CYP3A4, CYP3A5, TCF1, UGT1A1, UGT1A9, irinotecan, and neoplasms.

Irinotecan and CYP3A5 Genotype — PK

Submitted by: Mark J. Ratain, MD involving CYP3A5, irinotecan, and neoplasms.

Pharmacokinetics of irinotecan in cancer patients — PK

Submitted by: Howard McLeod, PharmD involving ABCB1, ABCC1, ABCC2, ABCG2, CES1, CES2, CYP3A4, CYP3A5, UGT1A1, ABCC1, irinotecan, and neoplasms.

Protein expression of genes in the irinotecan pathway — FA

Submitted by: Howard McLeod, PharmD involving ABCB1, ABCC1, ABCC2, ABCG2, ABCG2, ABCG2, CES2, CYP3A4, CYP3A5, TOP1, UGT1A1, irinotecan, and neoplasms.

Pathways
- Irinotecan Pathway
- Irinotecan Pathway (Cancer)

Legend
- phenotype data available
- genotype data available
- literature annotations available

Literature Annotations

Related Genes	Relationship	Details
ABCB1	CO PK FA GN	
ABCC1	FA GN	
ABCG2	FA GN	
BCHE	FA	

Pathway: SN-38G

http://www.pharmgkb.org/search/pathway/irinotecan

SN-38 Glucuronidation and UGT1A1 genotype
UGT1A1 (TA)n genotype - phenotype relationship in human livers. Liver microsomes were phenotyped for SN-38 glucuronidation rates in each liver with a single experiment performed in triplicate. Bars show the mean value of SN-38 glucuronidation rates in each group
Phenotype File: "Irinotecan Clinical Data"

PMID: 12464801

Pathway: Irinotecan-metabolite

http://www.pharmgkb.org/search/p...

From Santos et al, (2000) Clin Cancer Res 6: 2012-2020.

PMID: 10815927

Irinotecan *Link to Irinotecan drug page*

Different strategies and formats can be used for exchanging different types of data sets. An XML schema has been developed for genomic data sets. Rules and privileges can be set for different users for security purposes. In a pharmacogenetics and pharmacogenomics knowledge base, patient privacy can be well protected.

4.3.1 Querying System

The popularity of Google-style searches leads users to expect that the entire contents of a database can be searched for the occurrence of short text phrases. Many databases, including PharmGKB, index their contents for full-text search (using the Lucene [26] text search indexing engine). The challenge in this task is to present the results (which will be a mixture of web pages, data, publications and other types) to the user in a ranked and orderly way. Lucene ranks all results based on parameters and preferences that can be set by the developer. It also supports wildcards, boosting, approximate matches, and neighborhood searches (e.g. within 10 words).

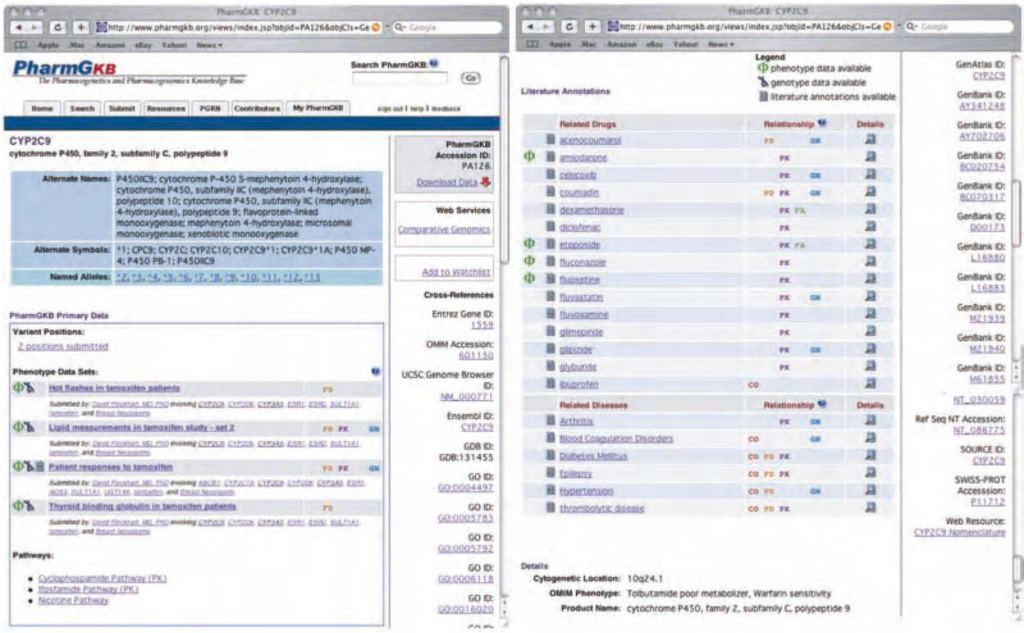
Most database users do not use powerful search features, but instead prefer either Google-style text searches or queries predefined on the basis of templates. From a set of use-case scenarios from pharmacogenomics researchers, we have identified a collection of commonly occurring queries and coded these on a single web page. Each predefined query has two or three "fill in the blank" selections and a submission button. The results are then formatted in a manner that is sensitive to the nature of the question – an advantage of knowing the query that the user has selected.

4.3.2 Visualization and Browsing

In order to present a consistent predictable picture of the database to the user, it is important to use a simple set of web design principles. The main objects in pharmacogenomics are genes, drugs, diseases, pathways and submission group. In PharmGKB, the gene, drug and disease pages have been designed to have a similar look and feel. Data for which PharmGKB is the primary home is boxed to draw attention. Literature annotations are included at the bottom of each page, and with links among genes, drugs and diseases. External links

←

Figure 4 Irinotecan pathways in liver, blood, bile and intestinal compartments. Genes are represented by ellipsoids, drugs and metabolites by rectangles; arrows for which there is supporting data have golden heads. All graphical objects link to gene, drug (B), primary data (C) or metabolite (D) pages to allow browsing. All PharmGKB pathways are constructed using similar graphical elements.


PharmGKB: CYP2C9
http://www.pharmgkb.org/views/index.jsp?objId=PA126&objCls=Ge
Google
Apple Mac Amazon eBay Yahoo! News
PharmGKB
The Pharmacogenetics and Pharmacogenomics Knowledge Base
Search PharmGKB:
Go
Home
Search
Submit
Resources
PGRN
Contributors
My PharmGKB
sign out | help | feedback

CYP2C9
cytochrome P450, family 2, subfamily C, polypeptide 9

Alternate Names:	P450IIC9; cytochrome P-450 S-mephenytoin 4-hydroxylase; cytochrome P450, subfamily IIC (mephenytoin 4-hydroxylase), polypeptide 10; cytochrome P450, subfamily IIC (mephenytoin 4-hydroxylase), polypeptide 9; flavoprotein-linked monooxygenase; mephenytoin 4-hydroxylase; microsomal monooxygenase; xenobiotic monooxygenase
Alternate Symbols:	*1; CPC9; CYP2C; CYP2C10; CYP2C9*1; CYP2C9*1A; P450 MP-4; P450 PB-1; P450IIC9
Named Alleles:	*2, *3, *4, *5, *6, *7, *8, *9, *10, *11, *12, *13

PharmGKB Accession ID: PA126
Download Data

Web Services
Comparative Genomics

Add to Watchlist

PharmGKB Primary Data

Variant Positions:
2 positions submitted

Phenotype Data Sets:

ɸ Hot flashes in tamoxifen patients PD
Submitted by: David Flockhart, MD, PhD involving CYP2C8, CYP2D6, CYP3A4, ESR1, ESR2, SULT1A1, tamoxifen, and Breast Neoplasms

ɸ Lipid measurements in tamoxifen study - set 2 PD PK GN
Submitted by: David Flockhart, MD, PhD involving CYP2C8, CYP2D6, CYP3A4, ESR1, ESR2, SULT1A1, tamoxifen, and Breast Neoplasms

ɸ Patient responses to tamoxifen PD PK GN
Submitted by: David Flockhart, MD, PhD involving ABCB1, CYP2C19, CYP2C8, CYP2D6, CYP3A4, ESR1, NQO1, SULT1A1, UGT1A4, tamoxifen, and Breast Neoplasms

ɸ Thyroid binding globulin in tamoxifen patients PD
Submitted by: David Flockhart, MD, PhD involving CYP2C8, CYP2D6, CYP3A4, ESR1, ESR2, SULT1A1, tamoxifen, and Breast Neoplasms

Pathways:

- Cyclophosphamide Pathway (PK)
- Ifosfamide Pathway (PK)
- Nicotine Pathway

Cross-References
Entrez Gene ID: 1559
OMIM Accession: 601130
UCSC Genome Browser ID: NM_000771
Ensembl ID: CYP2C9
GDB ID: GDB:131455
GO ID: GO:0004497
GO ID: GO:0005783
GO ID: GO:0005792
GO ID: GO:0006118
GO ID: GO:0016020

Legend
ɸ phenotype data available
genotype data available
literature annotations available

Literature Annotations

	Related Drugs	Relationship	Details
	acenocoumarol	PD GN	
ɸ	amiodarone	PK	
	celecoxib	PK GN	
	coumadin	PD PK GN	
	dexamethasone	PK FA	
	diclofenac	PK	
ɸ	etoposide	PK FA	
ɸ	fluconazole	PK	
ɸ	fluoxetine	PK	
	fluvastatin	PK GN	
	fluvoxamine	PK	
	glimepiride	PK	
	glipizide	PK GN	
	glyburide	PK	
	ibuprofen	CO	

	Related Diseases	Relationship	Details
	Arthritis	PK GN	
	Blood Coagulation Disorders	CO GN	
	Diabetes Mellitus	CO PD PK	
	Epilepsy	CO PD PK	
	Hypertension	CO PD GN	
	thrombolytic disease	CO PD PK	

Details
Cytogenetic Location: 10q24.1
OMIM Phenotype: Tolbutamide poor metabolizer, Warfarin sensitivity
Product Name: cytochrome P450, family 2, subfamily C, polypeptide 9

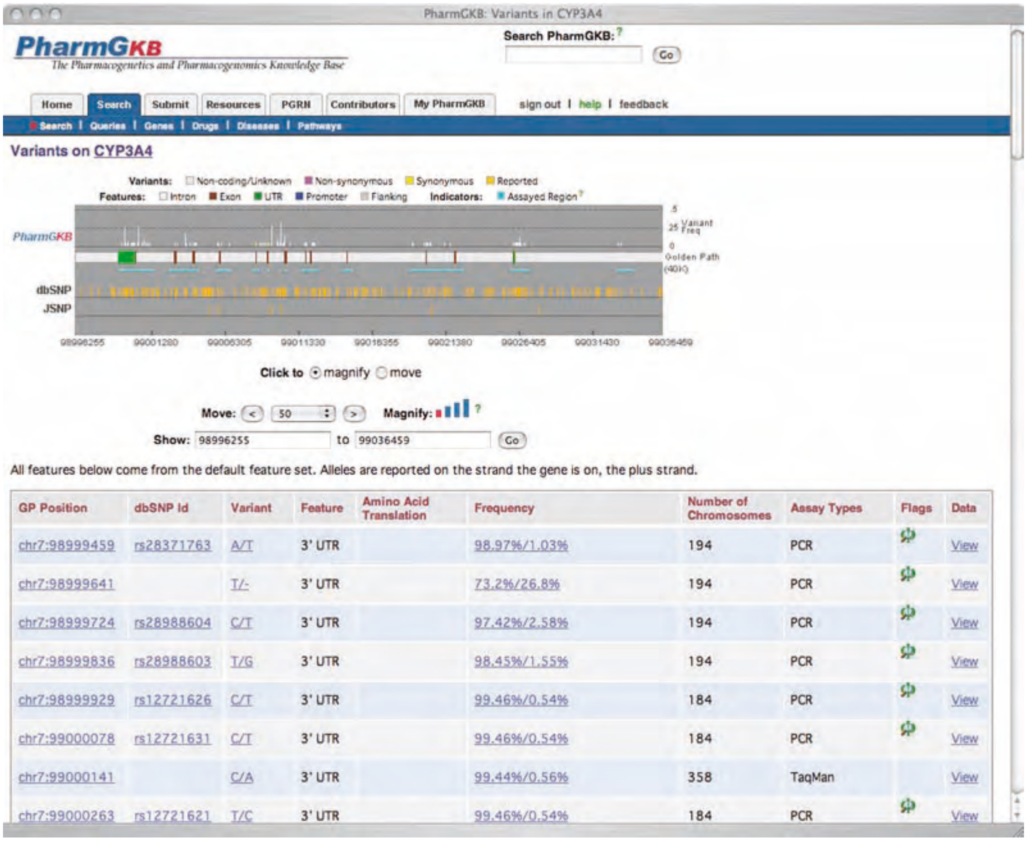
GenAtlas ID: CYP2C9
GenBank ID: AY341248
GenBank ID: AY702706
GenBank ID: BC020754
GenBank ID: BC070317
GenBank ID: D00173
GenBank ID: L16880
GenBank ID: L16883
GenBank ID: M21939
GenBank ID: M21940
GenBank ID: M61855
NT_030059
Ref Seq NT Accession: NT_086775
SOURCE ID: CYP2C9
SWISS-PROT Accession: P11712
Web Resource: CYP2C9 Nomenclature

Figure 5 Gene page for *CYP2C9*. Named alleles are shown on the top of the page. Primary data are shown in the boxed area (left panel). The far right has links to external sources. The bottom of the page (right panel) lists all curated relations between drugs and diseases, and indicates what category of knowledge is used to establish the relationship. All gene pages have a similar layout.

usually appear on the right panel. The current gene page for *CYP2C9* is shown in Figure 5.

A genotype browser is critical for visualization and analysis of genotype data. These browsers typically display a genomic segment, while indicating the location of polymorphisms. The PharmGKB browser allows for zooming, sliding and display of the gene structure (Figure 6). Below the browser is a table of polymorphisms displaying summary data and links, including (most significantly) links to any measured phenotypes of the listed genotypes.

Phenotype files in PharmGKB are presented first with a short textual summary including the genes, drugs and diseases studied and the categories of pharmacogenetic knowledge – clinical outcomes, pharmacodynamics and drug responses, pharmacokinetics and molecular and cellular functional assays. In the Details Section, each column is labeled by a controlled vocabulary to facilitate indexing and retrieval. Each row contains individual subject data which are linked to all genotypes for that subject via a unique identifier. All subject data is completely de-identified prior to submission to PharmGKB. An example is shown in Figure 7.


PharmGKB: Variants in CYP3A4
PharmGKB
The Pharmacogenetics and Pharmacogenomics Knowledge Base
Search PharmGKB: ?
Go
Home
Search
Submit
Resources
PGRN
Contributors
My PharmGKB
sign out | help | feedback
Search | Queries | Genes | Drugs | Diseases | Pathways
Variants: □ Non-coding/Unknown ■ Non-synonymous ■ Synonymous ■ Reported
Features: □ Intron ■ Exon ■ UTR ■ Promoter ■ Flanking Indicators: ■ Assayed Region ?
PharmGKB
dbSNP
JSNP
.5
.25 Variant Freq
0
Golden Path (40K)
98996255
99001280
99006305
99011330
99016355
99021380
99026405
99031430
99036459
Click to ● magnify ○ move
Move: < 50 >
Magnify: ?
Go

Variants on CYP3A4

Show: 98996255 to 99036459

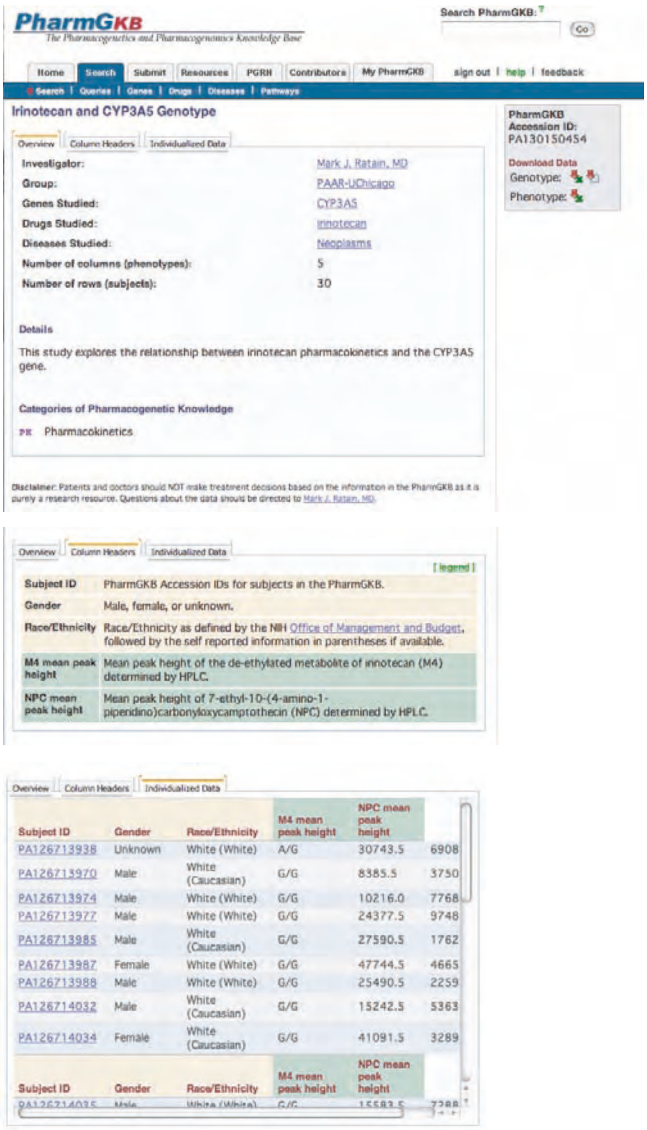
All features below come from the default feature set. Alleles are reported on the strand the gene is on, the plus strand.

GP Position	dbSNP Id	Variant	Feature	Amino Acid Translation	Frequency	Number of Chromosomes	Assay Types	Flags	Data
chr7:98999459	rs28371763	A/T	3' UTR		98.97%/1.03%	194	PCR	✓	View
chr7:98999641		T/-	3' UTR		73.2%/26.8%	194	PCR	✓	View
chr7:98999724	rs28988604	C/T	3' UTR		97.42%/2.58%	194	PCR	✓	View
chr7:98999836	rs28988603	T/G	3' UTR		98.45%/1.55%	194	PCR	✓	View
chr7:98999929	rs12721626	C/T	3' UTR		99.46%/0.54%	184	PCR	✓	View
chr7:99000078	rs12721631	C/T	3' UTR		99.46%/0.54%	184	PCR	✓	View
chr7:99000141		C/A	3' UTR		99.44%/0.56%	358	TaqMan		View
chr7:99000263	rs12721621	T/C	3' UTR		99.46%/0.54%	184	PCR	✓	View

Figure 6 Variant browser for *CYP3A4*. The genomic sequence is displayed with Golden Path numbering. Brown bands indicate the location of exons. SNPs are drawn as tick marks above the sequence bar, with height proportional to overall frequency and colored based on type of SNP (e.g. nonsynonymous). All SNPs are also displayed in the table below, with links to detailed frequency and population information. Haplotypes can be computed using web services at the top right.

4.3.3 **Privacy Protection**

Pharmacogenomics studies individual response to drugs and so individual data is critical. However, it is also critical to ensure the privacy of individuals from whom the genotype and phenotype data were collected [39, 40]. At PharmGKB, genotypic and phenotypic information associated with individual human subject is only available to registered users, who are associated with a *bona fide* research institution or company (or who are sponsored by other registered users). The most privacy-sensitive information has been de-identified or binned according to the method described by Lin and coworkers [39] in the database before it is released. Registration is free to the life science


PharmGKB
The Pharmacogenetics and Pharmacogenomics Knowledge Base
Search PharmGKB:
Go
Home
Search
Submit
Resources
PGRN
Contributors
My PharmGKB
sign out | help | feedback
Search | Queries | Genes | Drugs | Diseases | Pathways
Overview
Column Headers
Individualized Data
[legend]

Irinotecan and CYP3A5 Genotype

Investigator:	Mark J. Ratain, MD
Group:	PAAR-UChicago
Genes Studied:	CYP3A5
Drugs Studied:	irinotecan
Diseases Studied:	Neoplasms
Number of columns (phenotypes):	5
Number of rows (subjects):	30

Details

This study explores the relationship between irinotecan pharmacokinetics and the CYP3A5 gene.

Categories of Pharmacogenetic Knowledge

PK Pharmacokinetics

Disclaimer: Patients and doctors should NOT make treatment decisions based on the information in the PharmGKB as it is purely a research resource. Questions about the data should be directed to Mark J. Ratain, MD.

PharmGKB Accession ID: PA130150454

Download Data
Genotype:
Phenotype:

Subject ID	PharmGKB Accession IDs for subjects in the PharmGKB.
Gender	Male, female, or unknown.
Race/Ethnicity	Race/Ethnicity as defined by the NIH Office of Management and Budget, followed by the self reported information in parentheses if available.
M4 mean peak height	Mean peak height of the de-ethylated metabolite of innotecan (M4) determined by HPLC.
NPC mean peak height	Mean peak height of 7-ethyl-10-(4-amino-1-pipendino)carbonyloxycamptothecin (NPC) determined by HPLC.

Subject ID	Gender	Race/Ethnicity	M4 mean peak height	NPC mean peak height	
PA126713938	Unknown	White (White)	A/G	30743.5	6908
PA126713970	Male	White (Caucasian)	G/G	8385.5	3750
PA126713974	Male	White (White)	G/G	10216.0	7768
PA126713977	Male	White (White)	G/G	24377.5	9748
PA126713985	Male	White (Caucasian)	G/G	27590.5	1762
PA126713987	Female	White (White)	G/G	47744.5	4665
PA126713988	Male	White (White)	G/G	25490.5	2259
PA126714032	Male	White (Caucasian)	G/G	15242.5	5363
PA126714034	Female	White (Caucasian)	G/G	41091.5	3289
Subject ID	Gender	Race/Ethnicity	M4 mean peak height	NPC mean peak height	
PA126714035	Male	White (White)	G/G	15583.5	7288

Figure 7 Phenotype data display. In addition to view the phenotype data set from this page, phenotype and related genotype data can also be down loaded by clicking the red arrow on the top right corner. All phenotype data set have a similar layout.

research community, but one has to first provide an identity and agree to fully comply with the usage policies, which include the prohibition of any attempt to re-identify study subjects.

4.3.4 Data Exchange Strategy

PharmGKB has developed a general purpose XML schema for genotype data submission and exchange. This schema contains a superset of information required by dbSNP and GenBank and is shared publicly. The XML captures information about reference sequences, polymorphism location and type (e.g. SNP, deletion, insertion etc.), study samples, populations and methods for determining polymorphisms (e.g. sequencing, restriction fragment length polymorphism). It also allows for specification of species, copy numbers, and gene structure/annotations. It is available at http://www.pharmgkb.org/schema/index.html. Changes occur on a 6-months basis and are associated with a sufficiently timely notice to allow associated code to be modified. PharmGKB has participated in the public process of defining standards for exchange of SNPs, such as PML [61], and has ensured that emerging standards are consistent with our XML.

The availability of an XML schema with clear semantics allows curators to run a full syntactic and semantic validation on submitted files upon receipt of a submission. The validation process ranges from data type checks to "common-sense" checks, such as that base numbers fall within specified ranges and that key required data items are provided. This XML schema has provided a high-throughput processing of genotype submissions and posting on the web.

In addition to supporting bioinformatics-savvy submitters who use XML, it is important to have options for users with little bioinformatics expertise. For example, user-friendly web pages for data entry using Excel files are also available on PharmGKB. Laboratories without bioinformatics infrastructure can use the online submission option for their genotype data sets. These web submission pages are based on the data representation used in the XML schema for polymorphisms.

5 Analytic Tools for Pharmacogenomics

The field of pharmacogenomics is so broad that it is difficult to list the key analytic functionalities that are required. However, we can summarize the key analysis tasks in pharmacogenomics, and the types of tools that are used.

The first challenge in understanding how a drug-related phenotype depends on genetics is to identify the genes that are likely to be involved. This may involve the creation of "candidate gene" lists based on analysis of the published literature on the drug or phenotype. It may also involve examining pathways of drug metabolism or action. Any bioinformatics tools that can infer gene function, and relate a function to a drug response could be useful at this stage of pharmacogenomic analysis. The analysis of gene expression (e.g. using microarrays) can also be used to suggest which genes may be involved in the response to a drug. Presumably, similar proteomic capabilities will emerge in the future. See Part VIII of this book for information on how to analyze gene and protein function with bioinformatics methods.

The discovery of genetic variation in genes that may be involved in drug response can be accomplished by resequencing multiple individuals and looking for variation. These analyses typically use the Phred/Phrap suite of programs to find evidence of multiple bases at a single position in the genome [35]. Other methods include mining of expressed sequence tag (EST) databases to find evidence of sequence polymorphism that is not likely to be due to sequencing errors [13].

Once a polymorphism has been detected, it is necessary to determine if it is likely to be functionally significant. Methods for assessing the likely functional significance of SNPs include the SIFT program [71], the MutDB resource [44] and others [8,62,70].

Phenotypes related to drug response are extremely variable. The key analytic challenge for phenotypes is to define them precisely, and to collect and record them well. The organization of phenotypes into ontologies is a critical need, because this allows the relationship between different phenotypes to be defined, in order to support indexing, search and aggregation. For example, "blood pressure" is a general phenotype that can be assayed in many ways. If the detailed relationship between these assays is recorded, then different blood pressure studies can be combined appropriately to achieve higher statistical power.

The key analytic challenge for pharmacogenetics is the discovery of genotype–phenotype relationships. If a phenotype is controlled primarily by a single gene locus, then a plot of the phenotype versus the allele observed at that locus is often a bimodal or trimodal distribution, with each mode corresponding to a different allele. If, however, the phenotype is controlled by a large number of loci from different genes, the distribution is likely to be unimodal and broad. Methods for finding the key genetic loci associated with a particular phenotypic measurement are the domain of statistical genetics and have been the subject of many recent reviews [11, 49, 72] (see also Chapter 37).

Recently, there has been a move to "whole-genome" association studies in which the phenotype is correlated to SNPs collected from a sample of genomic loci across the whole genome. The genotype SNPs are usually selected to be "tag SNPs" in linkage disequilibrium with a large number of other SNPs locally [59,73]. Once regions are found to be correlated with the phenotype of interest, they can be studied in a refined manner by genotyping the local SNPs and searching for those that are likely to be functionally significant, using the methods mentioned above. One strategy involves the combination of family-based linkage analysis to focus attention on a small number of genomic regions, association analysis with SNPs to find those SNPs in these regions that are most correlated with the phenotype of interest and then expression analysis to see if any of the genes showing potentially relevant polymorphisms show an expression change in response to a drug challenge. More information on this topic can be found in Chapter 37. Multiple other experimental methods can be used to focus attention on regions of the genome for which a variety of evidence points to involvement in modulating the response to a drug.

6 Future Perspectives on Informatics for Pharmacogenetics/Pharmacogenomics

PharmGKB is the first public pharmacogenetics and pharmacogenomics database to provide information about genetic variations in humans and the phenotypic consequences of those variations. By providing original data in standardized formats, and web interfaces to retrieve integrated information and for analyzing data, PharmGKB is providing solutions to many of the challenges that pharmacogenetic and pharmcogenomic researchers brought to biomedical informatics.

One of the major goals of pharmacogenomics is to predict drug response in individuals based on their genetic profile in order to provide the most effective individualized medicine. In the future, patients will be stratified based on genetic tests for their ability to respond to a therapeutic agent and their possibility of having adverse reactions. The process of drug development will also benefit from the ability to identify targeted patient populations with a high likelihood of success early in the process, while avoiding unexpected late failures. As the field increases its inventory of high-impact data sets and useful analytical tools, it should help catalyze the use of patient genetic information in medical practice.

Acknowledgments

This work is supported by the NIH/NIGMS Pharmacogenetics Research Network and Database (U01GM61374).

References

1 International HapMap Consortium. 2004. Integrating ethics and science in the International HapMap Project. Nat. Rev. Genet. **5**: 467–75.

2 International HapMap Consortium. 2003. The International HapMap Project. Nature **426**: 789–96.

3 Altman, R. B., D. A. Flockhart, S. T. Sherry, D. E. Oliver, D. L. Rubin and T. E. Klein. 2003. Indexing pharmacogenetic knowledge on the World Wide Web. Pharmacogenetics **13**: 3–5.

4 Bairoch, A., R. Apweiler, C. H. Wu, et al. 2005. The Universal Protein Resource (UniProt). Nucleic Acids Res **33**: D154–9.

5 Benson, D. A., I. Karsch-Mizrachi, D. J. Lipman, J. Ostell and D. L. Wheeler. 2005. GenBank. Nucleic Acids Res **33**: D34–8.

6 Birney, E., T. D. Andrews, P. Bevan, et al. 2004. An overview of Ensembl. Genome Res. **14**: 925–8.

7 Brown, S. D., Jr. and F. J. Landry. 2001. Recognizing, reporting, and reducing adverse drug reactions. South Med. J. **94**: 370–3.

8 Clifford, R. J., M. N. Edmonson, C. Nguyen and K. H. Buetow. 2004. Large-scale analysis of non-synonymous coding region single nucleotide polymorphisms. Bioinformatics **20**: 1006–14.

9 D'Argenio, D. Z. and A. Schumitzky. 1979. A program package for simulation and parameter estimation in pharmacokinetic systems. Comput. Programs Biomed. **9**: 115–34.

10 Dahlquist, K. D., N. Salomonis, K. Vranizan, S. C. Lawlor and B. R. Conklin. 2002. GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. Nat. Genet. **31**: 19–20.

11 Daly, A. K. 2003. Candidate gene case-control studies. Pharmacogenomics **4**: 127–39.

12 Daly, A. K. 2004. Pharmacogenetics of the cytochromes P450. Curr. Top. Med. Chem. **4**: 1733–44.

13 Dantec, L. L., D. Chagne, D. Pot, et al. 2004. Automated SNP detection in expressed sequence tags: statistical considerations and application to maritime pine sequences. Plant Mol. Biol. **54**: 461–70.

14 Dantzer, J., C. Moad, R. Heiland and S. Mooney. 2005. MutDB services: interactive structural analysis of mutation data. Nucleic Acids Res. **33**: W311–4.

15 Deshpande, N., K. J. Addess, W. F. Bluhm, et al. 2005. The RCSB Protein Data Bank: a redesigned query system and relational database based on the mmCIF schema. Nucleic Acids Res. **33**: D233–7.

16 Diehn, M., G. Sherlock, G. Binkley, et al. 2003. SOURCE: a unified genomic resource of functional annotations, ontologies, and gene expression data. Nucleic Acids Res. **31**: 219–23.

17 Fenton, S. and M. Williams. 2005. Getting to know PubMed: an overview. J. Ahima **76**: 60A–D.

18 Frezal, J. 1998. Genatlas database, genes and development defects. C. R. Acad. Sci. III **321**: 805–17.

19 Giovannetti, E., V. Mey, S. Nannizzi, G. Pasqualetti, L. Marini, M. Del Tacca and R. Danesi. 2005. Cellular and pharmacogenetics foundation of synergistic interaction of pemetrexed and gemcitabine in human non-small cell lung cancer cells. Mol. Pharmacol. **68**: 110–8.

20 Gkoutos, G. V., E. C. Green, A. M. Mallon, J. M. Hancock and D. Davidson. 2005. Using ontologies to describe mouse phenotypes. Genome Biol. **6**: R8.

21 Guttmacher, A. E. and F. S. Collins. 2002. Genomic medicine – a primer. N. Engl. J. Med. **347**: 1512–20.

22 Hahn, L. W., M. D. Ritchie and J. H. Moore. 2003. Multifactor dimensionality reduction software for detecting gene–gene and gene–environment interactions. Bioinformatics **19**: 376–82.

23 Hakonarson, H., S. Thorvaldsson, A. Helgadottir, et al. 2005. Effects

of a 5-lipoxygenase-activating protein inhibitor on biomarkers associated with risk of myocardial infarction: a randomized trial. J. Am. Med. Ass. **293**: 2245–56.

24 Hall, A. M. and M. R. Wilkins. 2005. Warfarin: a case history in pharmacogenetics. Heart **91**: 563–4.

25 Hamosh, A., A. F. Scott, J. S. Amberger, C. A. Bocchini and V. A. McKusick. 2005. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic Acids Res. **33**: D514–7.

26 Hatcher, E. and O. Gospodnetic. 2004. *Lucene in Action*. Manning, Greenwich, CT.

27 Helgadottir, A., S. Gretarsdottir, D. St Clair, et al. 2005. Association between the gene encoding 5-lipoxygenase-activating protein and stroke replicated in a Scottish population. Am. J. Hum. Genet. **76**: 505–9.

28 Helgadottir, A., A. Manolescu, G. Thorleifsson, et al. 2004. The gene encoding 5-lipoxygenase activating protein confers risk of myocardial infarction and stroke. Nat. Genet. **36**: 233–9.

29 Hider, S. L., C. Buckley, A. J. Silman, D. P. Symmons and I. N. Bruce. 2005. Factors influencing response to disease modifying antirheumatic drugs in patients with rheumatoid arthritis. J. Rheumatol. **32**: 11–16.

30 Hucka, M., A. Finney, H. M. Sauro, et al. 2003. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics **19**: 524–31.

31 Kalow, W. 2004. Human pharmacogenomics: the development of a science. Hum. Genomics **1**: 375–80.

32 Kalow, W. 1962. *Pharmacogenetics: Heredity and the Response to Drugs*. Saunders, Philadelphia, PA.

33 Kanehisa, M., S. Goto, S. Kawashima, Y. Okuno and M. Hattori. 2004. The KEGG resource for deciphering the genome. Nucleic Acids Res. **32**: D277–80.

34 Kent, W. J., C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle, A. M. Zahler and D. Haussler. 2002. The human genome browser at UCSC. Genome Res. **12**: 996–1006.

35 Lee, W. H. and V. B. Vega. 2004. Heterogeneity detector: finding heterogeneous positions in Phred/Phrap assemblies. Bioinformatics **20**: 2863–4.

36 Lennard, L., I. J. Lewis, M. Michelagnoli and J. S. Lilleyman. 1997. Thiopurine methyltransferase deficiency in childhood lymphoblastic leukaemia: 6-mercaptopurine dosage strategies. Med. Pediatr. Oncol. **29**: 252–5.

37 Lennard, L., J. S. Lilleyman, J. Van Loon and R. M. Weinshilboum. 1990. Genetic variation in response to 6-mercaptopurine for childhood acute lymphoblastic leukaemia. Lancet **336**: 225–9.

38 Lennard, L., J. A. Van Loon and R. M. Weinshilboum. 1989. Pharmacogenetics of acute azathioprine toxicity: relationship to thiopurine methyltransferase genetic polymorphism. Clin. Pharmacol. Ther. **46**: 149–54.

39 Lin, Z., M. Hewett and R. B. Altman. 2002. Using binning to maintain confidentiality of medical data. Proc. AMIA Symp., 2002; 454–8.

40 Lin, Z., A. B. Owen and R. B. Altman. 2004. Genetics. Genomic research and human subject privacy. Science **305**: 183.

41 Lynch, T. J., D. W. Bell, R. Sordella, et al. 2004. Activating mutations in the epidermal growth factor receptor underlying responsiveness of non-small-cell lung cancer to gefitinib. N. Engl. J. Med. **350**: 2129–39.

42 Maglott, D., J. Ostell, K. D. Pruitt and T. Tatusova. 2005. Entrez Gene: gene-centered information at NCBI. Nucleic Acids Res. **33**: D54–8.

43 Meyer, U. A. and U. M. Zanger. 1997. Molecular mechanisms of genetic polymorphisms of drug metabolism. Annu. Rev. Pharmacol. Toxicol. **37**: 269–96.

44 Mooney, S. D. and R. B. Altman. 2003. MutDB: annotating human variation with functionally relevant data. Bioinformatics **19**: 1858–60.

45 Motulsky, A. G. 1957. Drug reactions, enzymes, and biochemical genetics. J. Am. Med. Ass. **165**: 835–7.

46 Paez, J. G., P. A. Janne, J. C. Lee, et al. 2004. EGFR mutations in lung cancer: correlation with clinical response to gefitinib therapy. Science **304**: 1497–500.

47 Pinsonneault, J. and W. Sadee. 2003. Pharmacogenomics of multigenic diseases: sex-specific differences in disease and treatment outcome. AAPS PharmSci. **5**: E29.

48 Pruitt, K. D., T. Tatusova and D. R. Maglott. 2005. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Res. **33**: D501–4.

49 Rannala, B. 2001. Finding genes influencing susceptibility to complex diseases in the post-genome era. Am. J. Pharmacogenomics **1**: 203–21.

50 Rebhan, M., V. Chalifa-Caspi, J. Prilusky and D. Lancet. 1997. GeneCards: integrating information about genes, proteins and diseases. Trends Genet. **13**: 163.

51 Redfern, W. S., I. D. Wakefield, H. Prior, C. E. Pollard, T. G. Hammond and J. P. Valentin. 2002. Safety pharmacology – a progressive approach. Fundam. Clin. Pharmacol. **16**: 161–73.

52 Ritchie, M. D., L. W. Hahn, N. Roodi, L. R. Bailey, W. D. Dupont, F. F. Parl and J. H. Moore. 2001. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. Am. J. Hum. Genet. **69**: 138–47.

53 Roots, I., T. Gerloff, C. Meisel, et al. 2004. Pharmacogenetics-based new therapeutic concepts. Drug Metab. Rev. **36**: 617–38.

54 Rose, A. D. 2004. Pharmacogenetics and drug development: the path to safer and more effective drugs. Nat. Rev. Genet. **5**: 645–656.

55 Rubin, D. L., M. Carrillo, M. Woon, J. Conroy, T. E. Klein and R. B. Altman. 2004. A resource to acquire and summarize pharmacogenetics knowledge in the literature. Medinfo **2004**: 793–7.

56 Sachidanandam, R., D. Weissman, S. C. Schmidt, et al. 2001. A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. Nature **409**: 928–33.

57 Spear, B. B., M. Heath-Chiozzi and J. Huff. 2001. Clinical application of pharmacogenetics. Trends Mol. Med. **7**: 201–4.

58 Spellman, P. T., M. Miller, J. Stewart, et al. 2002. Design and implementation of microarray gene expression markup language (MAGE-ML). Genome Biol. **3**: RESEARCH0046.

59 Stram, D. O. 2004. Tag SNP selection for association studies. Genet. Epidemiol. **27**: 365–74.

60 Stryke, D., C. C. Huang, M. Kawamoto, et al. 2003. SNP analysis and presentation in the Pharmacogenetics of Membrane Transporters Project. Pac Symp. Biocomput. **8**: 535–47.

61 Sugawara, H., H. Mizushima, T. Kano, Y. Shigemoto, et al. 2004. Polymorpism Markup Language (PML) for the interoperability of data on SNPs and other sequence variations. In Proc. Genome Informatics Workshop.

62 Taylor, N. E. and E. A. Greene. 2003. PARSESNP: a tool for the analysis of nucleotide polymorphisms. Nucleic Acids Res. **31**: 3808–11.

63 Thorisson, G. A. and L. D. Stein. 2003. The SNP Consortium website: past, present and future. Nucleic Acids Res. **31**: 124–7.

64 Vogel, F. 1959. Moderne Probleme der Humangenetik. Ergebn. Inn. Med. Kinderheilkd. **12**: 52–125.

65 Wain, H. M., M. J. Lush, F. Ducluzeau, V. K. Khodiyar and S. Povey. 2004. Genew: the Human Gene Nomenclature Database, 2004 updates. Nucleic Acids Res. **32**: D255–7.

66 Weinshilboum, R. and L. Wang. 2004. Pharmacogenomics: bench to bedside. Nat. Rev. Drug Discov. **3**: 739–48.

67 Weinshilboum, R. M., D. M. Otterness and C. L. Szumlanski.

1999. Methylation pharmacogenetics: catechol *O*-methyltransferase, thiopurine methyltransferase, and histamine *N*-methyltransferase. Annu. Rev. Pharmacol. Toxicol. **39**: 19–52.

68 WEINSHILBOUM, R. M. AND S. L. SLADEK. 1980. Mercaptopurine pharmacogenetics: monogenic inheritance of erythrocyte thiopurine methyltransferase activity. Am. J. Hum. Genet. **32**: 651–62.

69 WILLIAMS, R. J. 1956. *Biochemical Individuality*. Wiley, New York, NY.

70 XI, T., I. M. JONES AND H. W. MOHRENWEISER. 2004. Many amino acid substitution variants identified in DNA repair genes during human population screenings are predicted to impact protein function. Genomics **83**: 970–9.

71 XIONG, M., J. ZHAO AND E. BOERWINKLE. 2003. Haplotype block linkage disequilibrium mapping. Front. Biosci. **8**: a85–93.

72 XU, J., D. G. WIESCH AND D. A. MEYERS. 1998. Genetics of complex human diseases: genome screening, association studies and fine mapping. Clin. Exp. Allergy **28 (Suppl. 5)**: 1–5; discussion 26–8.

73 ZHANG, K., Z. S. QIN, J. S. LIU, T. CHEN, M. S. WATERMAN AND F. SUN. 2004. Haplotype block partitioning and tag SNP selection using genotype data and their applications to association studies. Genome Res. **14**: 908–16.

74 ZHAO, T., L. W. CHANG, H. L. MCLEOD AND G. D. STORMO. 2004. PromoLign: a database for upstream region analysis and SNPs. Hum. Mutat. **23**: 534–9.

40
Evolution of Drug Resistance in HIV

Niko Beerenwinkel, Kirsten Roomp, and Martin Däumer

1 Introduction

Evolution is the cornerstone of modern biology, because it provides a unifying principle for understanding diverse biological phenomena and for the quantitative analysis of molecular data. Evolutionary concepts also become increasingly important in medicine as witnessed, for example, by the progress in finding the genetic basis of complex diseases (Chapter 38) or by assessing polymorphisms in the human genome that are associated with drug effectiveness and tolerance (pharmacogenomics, Chapter 39). Evolutionary and population biology methods play a key role in understanding and controlling infectious diseases both within and among individuals [64]. The primary examples are infectious bacteria (Chapter 41) and viruses. In this chapter, we investigate the evolution of HIV. HIV is an attractive model system for evolutionary studies due to its short genome, large population size and high genetic diversity. The extreme replication dynamics of HIV allow for observing significant evolutionary changes over time. Understanding the evolution of HIV is necessary for controlling the spread of the AIDS pandemic, for developing effective therapies and vaccines, and for managing drug resistance.

The development of drug resistance is a major obstacle to the successful treatment of HIV infection. The evolutionary dynamics of HIV facilitate its escape from the selective pressure exerted by the human immune system and by combination drug therapy. This chapter presents computational methods for the design of optimal antiretroviral therapies against drug-resistant HIV strains. The focus is on using genomic information, i.e. the nucleotide sequences of the viral drug targets, to guide treatment decisions. Since every patient carries a unique virus population, the presented methods support individualized therapies, rather than the traditional one-drug-for-all paradigm. As opposed to the individual treatment factors based on human genetic variation (Chapters 38 and 39), here the rationale for unique treatment choices arises in the first place from the genetic variation in the infectious pathogen.

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

Thus, we present computational methods for a specific case of personalized medicine.

Section 2 reviews the biomedical background of HIV infection, antiretroviral therapy, and drug resistance. In Section 3, we discuss machine learning and statistical methods which are used to predict phenotypic drug resistance, as measured *in vitro,* from the viral genotype. The development of drug resistance is the subject of Section 4. We present methods for learning mutational pathways which the virus takes in order to escape from drug pressure and we discuss the concept of the genetic barrier to resistance. Section 5 focuses on estimating the *in vivo* effect of drug combinations conditioned on the baseline viral genotype. The described techniques draw on the results of previous sections and directly address the problem of selecting optimal drug combinations. In Section 6, the effect of immune pressure on the evolution of the virus is studied. We discuss some open problems in Section 7 and list related web resources in the final Section 8.

2 Biomedical Background

2.1 Biology of HIV

2.1.1 Epidemiology of HIV/AIDS

AIDS is one of the most serious infectious diseases having ever affected humankind. An estimated 40 million people are currently suffering from this disease [105]. Its mortality rate is close to 100% and resulted in more than 3 million AIDS-related deaths in 2004. In the same year, 5 million people became infected with the relevant pathogen, the HIV, and of these 95% live in developing countries. Despite the existence of controversial theories for explaining the origin of HIV, there is wide agreement that the virus entered the human population in a zoonotic transmission from African nonhuman primates, which were infected with the closely related simian immunodeficiency viruses (SIVs) [44]. It is estimated that HIV type 1 (HIV-1) and type 2 (HIV-2) were introduced into humans around 1930 and in the 1940s, respectively [58,61].

AIDS was first recognized in 1981 in the US. At that time several reports described an increase in the incidence of rare opportunistic infections such as *Pneumocystis carinii* pneumonia and mucosal candidiasis [43]. Common features among the affected patients were that they showed evidence of a general immune deficiency and that they were either homosexual men or intravenous drug abusers. Two years later HIV was isolated [1] and by the mid-1980s it became evident that two types of HIV (HIV-1 and -2), with slightly different genome architectures, were circulating in the human population.

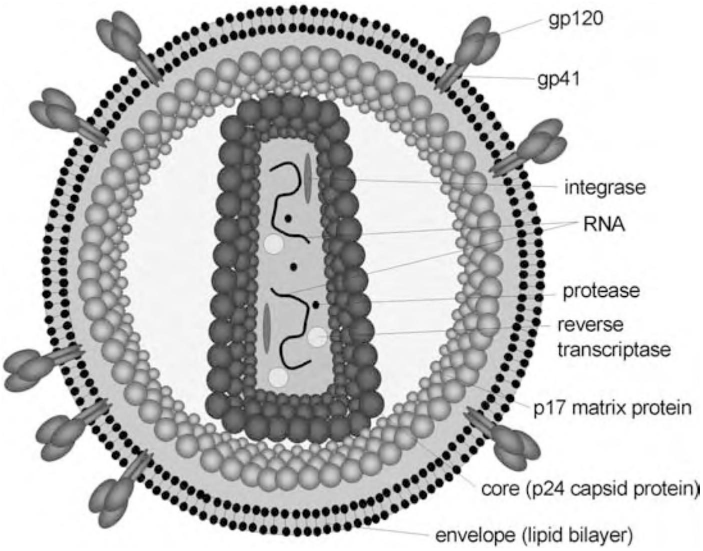


Figure 1 Schematic diagram of an HIV particle.

Both HIV-1 and -2 strains are divided into groups and subtypes. HIV-1 embraces the three genetically distinct groups M ("main"), N ("new") and O ("outlier"). Group M viruses, which cause 99% of HIV infections worldwide, are further subdivided into 11 genetically distinct subtypes, i.e. A1, A2, B, C, D, F1, F2, G, H, J and K [65]. Additionally, a major fraction of HIV-1 strains comprises intersubtype recombinants, designated "circulating recombinant forms" (CRF). The less virulent HIV-2 strains comprise six distinct phylogenetic lineages, i.e. subtypes A–F.

2.1.2 Structure, Genome and Replication Cycle

The HIV particle (virion) is roughly spherical and about 100–120 nm in diameter. Its outer envelope is composed of a lipid bilayer bearing numerous spikes (Figure 1). Each spike is composed of three heterodimers formed by glycoprotein 41 (gp41) and glycoprotein 120 (gp120) [40]. Beneath the outer envelope is a layer of matrix protein. The core of the virus particle has a hollow, truncated cone shape and is composed of another protein, p24, which contains the genetic material of the virus. Finally, inside the core there are two strands of RNA, consisting of about 9200 nucleotide bases, an integrase (p31), a protease (p10) and a reverse transcriptase (RT) (p51/66).

HIV is classified as a cytopathic retrovirus. Several characteristics differentiate retroviruses from other viral families. Their defining features are an RNA genome and the RT, an enzyme which facilitates the conversion of the RNA into DNA. HIV's genome consists of nine genes. Three genes, *gag*, *pol*

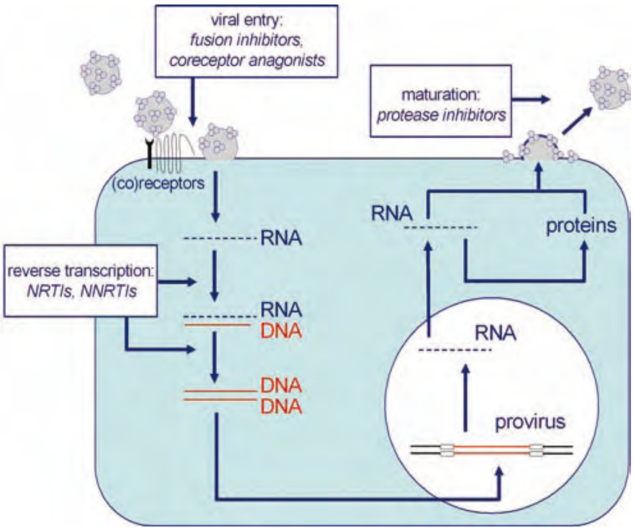


Figure 2 HIV replication cycle and targets of antiretroviral drug therapy.

and *env,* are common to all other retroviruses, the remaining six, the so-called accessory genes, *vif*, *vpr*, *tat*, *rev*, *nef* and *vpu*, are unique to HIV and SIV.

The life cycle of HIV can be divided into six steps: (i) binding to the target cell, (ii) penetration into the cell, (iii) reverse transcription, (iv) integration into the host's genome, (v) replication and (vi) budding of new virions (Figure 2). Gp120 facilitates binding and gp41 enables fusion of the viral envelope with the cell membrane. The primary host receptor for gp120 is CD4. CD4 receptors are most highly abundant on helper T cells, making these cells particularly susceptible to HIV infection. However, macrophages, monocytes, dendritic cells, Langerhans cells, hematopoietic stem cells, certain rectal-lining cells, and microglial cells are also susceptible. Subsequent to the initial binding of gp120 to CD4, binding to a cellular coreceptor, such as CCR5 or CXCR4, follows. Finally, binding to gp41 occurs enabling the penetration of the virus into the target cell.

Once inside the cell, the virus sheds its coat and transcribes its RNA into DNA. The resulting DNA is transported to the host cell nucleus and is integrated into the host cell's genome, forming what is referred to as the provirus. The virus can now remain dormant for an undetermined time period, the so-called latent phase, during which the viral genes are not expressed. Activation of the provirus is marked by the beginning of the transcription of HIV's structural genes and the formation of new virions. Transcription produces mRNA, which is translated into viral proteins, as well as single-stranded RNA (ssRNA), which is inserted into all new virus particles. The virus assembles into a new virion, which will bud out of the cell in order to enter new cells. The host cell membrane is modified by the insertion of gp41 and gp120. The viral ssRNA and core proteins assemble beneath the modified membrane, and, while budding, acquire the modified host plasma membrane as their envelope. Finally, the viral protease is required to process the precursor Gag–Pol polyproteins into mature HIV particles.

2.1.3 Basic Immunology and Course of Infection

HIV is able to infect several human cell types, but the most severe damage appears to result from the infection of cells that are of central importance to the immune system. The human immune system fights foreign invaders, removes dead and damaged cells, and destroys mutant and cancerous cells. The immune system is capable of fighting pathogens to which it has never been exposed employing a number of different cell types, which are all categorized as lymphocytes. B and T cells are two of the types of lymphocytes that recognize foreign substances or nonself. B cells produce and secrete antibodies in response to an antigen [76]. The three major types of T cells are cytotoxic or killer T cells (CTLs), suppressor T cells and helper T cells. CTLs eliminate virus-infected cells and are responsible for recovery from a viral infection. Suppressor T cells or cytotoxic T cells suppress the immune response after the antigen is eliminated. Helper T cells alert the immune system to antigens and signal other cells in the system to attack the antigen. Helper T cells do not kill cells, but interact with B cells and CTLs in order to help them attack foreign particles [98].

There are specialized receptors on the surface of each T cell to identify one of many millions of possible antigens that may invade the organism. Each T cell expresses a unique receptor that binds to the complementary antigen on the foreign particle to neutralize or destroy it. Killer and suppressor T cells carry the CD8 receptor (T suppressor cells are also called T8 cell) and the helper T cells (T4 cells) carry the CD4 receptor. Collectively, T8 and T4 cells regulate the body’s immune response to foreign antigens [98].

The primary infection elicits a rapid increase in CTL cells. It is estimated that approximately half of all T lymphocytes are involved. Rapid CTL cell division occurs over the first 2- to 3-week period. CTL levels peak and HIV-specific antibodies appear in the blood, leading to a drop in viral load, after which it is difficult to isolate the virus. This phase, referred to as clinical latency, is characterized by low viral replication and by a slow but constant decrease of the number of CD4$^+$ cells [36, 47].

The CTL response is essential for controlling the virus [67]. A strong CTL response correlates with low plasma viremia (amount of free virus in the blood) and a prolonged asymptomatic stage. The type of human leukocyte antigen (HLA, see Section 6), a kind of peptidic alarm system on cells, affects the effectiveness of the CTL response, as CTLs can only recognize an epitope bound to an HLA class I molecule.

CTLs have multiple antiviral mechanisms and it is currently unclear which of these mechanisms is most important for the control of viral infections. CTLs have the ability to lyse infected cells and to produce cytokines at sites of viral replication. Cytokines are low-molecular-weight soluble proteins that are produced in response to an antigen. They function as chemical messengers regulating the innate and adaptive immune systems (tumor necrosis factor-α, interleukin-1). Cytokines affect viral replication through their influence on T helper cell activation and proliferation. Certain chemokines, a group of cytokines, such as MIP-1α, MIP-1β and RANTES, are produced that suppress HIV replication by competing for or down-regulating the CCR5 coreceptor. However, at some point of the infection CTLs become incapable of controlling the virus. A possible factor contributing to this failure is the inability of other T cells to help. The immune system is able to reduce the number of viruses initially. Consequently, CTL-resistant HIV mutants are selected. These mutants are not easily recognized by HLA molecules. Since only those virions capable of escape survive and reproduce, eventually, the entire viral population consists of CTL escape mutants (Section 6). At this point the immune system is completely defenseless against the virus and HIV can freely reproduce, thereby further impairing the immune system. As a result, the increasing damage to the immune system leaves the infected individual susceptible to opportunistic infections.

2.2 Antiretroviral Therapy

2.2.1 Antiretroviral Drugs

The aim of antiretroviral therapy in HIV infection is to reduce the amount of replicating virus to as low a level as possible, thereby preventing the infection of new cells and further damage to the immune system. There are now 18 different drugs from four distinct drug classes in widespread use (Table 1). The first antiretroviral agent to become commercially available was zidovudine in 1987, followed by didanosine and zalcitabine in 1993. These drugs belong to the class of nucleoside analog RT inhibitors (NRTIs), which act on the RT (Figures 1 and 2). These drugs are analogues of thymidine or cytosine and are modified in the $3'$ position of the ribose molecule. NRTIs can be considered as prodrugs as they must be phosphorylated to become active. The incorporation of these drugs into newly synthesized DNA leads to premature chain termination as the drugs do not provide the $3'$ hydroxyl group to form the bond with the next nucleotide.

A dramatic decline in clinical progression of HIV disease and HIV-related deaths followed the introduction of protease inhibitors (PIs) in 1996. These

Table 1 Approved antiretroviral drugs in the four drug classes: NRTIs, NNRTIs, PIs and EIs (see text for details)

Drug class	Generic name	Abbreviation	Trade name
NRTI	zidovudine	ZDV	Retrovir
	didanosine	ddI	Videx
	zalcitabine	ddC	Hivid
	stavudine	d4T	Zerit
	lamivudine	3TC	Epivir
	abacavir	ABC	Ziagen
	tenofovir	TFV	Viread
	emtricitabin	FTC	Emtriva
NNRTI	nevirapine	NVP	Viramine
	delaviridine	DLV	Rescriptor
	efavirenz	EFV	Sustiva
PI	saquinavir	SQV	Invirase, Fortovase
	ritonavir	RTV	Norvir
	indinavir	IDV	Crixivan
	nelfinavir	NFV	Viracept
	lopinavir	LPV	Kaletra
	atazanavir	ATV	Reyataz
EI	enfuvirtide	ENF	Fuzeon

compounds act on the HIV protease enzyme, preventing the cleavage of essential viral precursor proteins. PIs are peptide-like molecules that mimic the Gag–Pol polyprotein and compete against it for the enzyme. The competitive inhibition of the viral protease leads to production of immature, noninfectious virus particles. Although PI therapy does not prevent the destruction of already infected cells, a further spread of infection into uninfected cell populations is prevented.

The nonnucleoside RT inhibitors (NNRTIs) are the third class of drugs currently available for treating HIV infection. NNRTIs also act on the RT enzyme, but differ in structure and mechanism of action from the NRTIs. The binding of an NNRTI close to the active site of the RT leads to a conformational change of the active site of the enzyme, and consequently to a reduced binding affinity for the RT's natural substrates, i.e. the nucleotides.

The fourth drug class comprises the so-called entry inhibitors (EIs), acting at the stage of viral attachment or host cell penetration. The only drug approved so far within this class is the fusion inhibitor enfuvirtide. Coreceptor antagonists are likely to enter phase III clinical trials soon.

The current standard of care is to treat HIV-infected individuals using a combination of at least three drugs out of two drug classes. This is commonly referred to as "highly active antiretroviral therapy" (HAART). Unfortunately, long-term use of antiretroviral treatment is associated with several limitations and drawbacks. Currently available antiretroviral drugs are unable to completely eradicate HIV, and virus continues to reside and to replicate latently in reservoirs. As a result, most patients will be on antiretroviral therapy for the rest of their lives, a situation that often causes continuous severe side-effects due to drug toxicity. Additionally, poor adherence to drug regimens has led to an increased rate of HIV drug resistance, resulting in viral strains that have reduced sensitivity to drug treatment [75].

2.2.2 Drug Resistance

The genetic and molecular basis of drug resistance is the enormous viral turnover coupled with an extremely high replication error rate, caused by the RT which lacks a proofreading mechanism. Mathematical modeling suggests that every possible point mutation in the viral genome may occur more than 10 000 times in an infected person per day [21]. Thus it is likely that mutants that are resistant to a single drug may naturally exist, whereas resistance to a drug combination requiring three or more specific mutations seems to be much more unlikely to preexist.

Resistance to antiretroviral drugs emerges when viral replication continues in the presence of selective drug pressure. For some drugs, such as the NRTI lamivudine and all NNRTIs, one point mutation is sufficient to induce high-level resistance, whereas other drugs like zidovudine and the PIs require the accumulation of multiple mutations to reach a high level of resistance. Thus, resistance to therapies consisting of combinations of different drugs develops in a gradual and stepwise manner (Section 4).

Single mutations or mutational patterns can result in resistance to an entire drug class. Therefore, a strain, which has emerged under the selective pressure of a specific drug, may be resistant not only to that drug, but also to other drugs from the same class that have never been applied. This phenomenon is referred to as cross-resistance. Cross-resistance affects almost all of the currently available drugs to varying degrees. Thus, resistance to one antiretroviral agent will affect the choice of other drugs from the same class.

2.3 Resistance Testing

Resistance to antiretroviral drugs can be measured using either genotypic or phenotypic assays. Genotypic assays detect mutations known to cause drug resistance. Phenotypic assays are drug susceptibility assays in which the virus is cultured in the presence of serial dilutions of an inhibitory drug. Both genotypic and phenotypic assays use HIV-1 RNA extracted from the patient's blood plasma for further testing, since this virus represents the phylogenetically most recent step in the viral evolution that is happening inside the patient.

2.3.1 Genotypic Resistance Testing

Genotypic resistance testing is generally performed by sequencing the drugs' target genes in order to detect mutations that confer drug resistance. The clinical usefulness of genotypic testing has been demonstrated in several prospective randomized trials [20, 26, 33]. Genotypic testing is used more commonly than phenotypic testing because of its lower cost, wider availability and shorter turnaround time. Furthermore, genotyping provides early evidence of drug resistance within a virus population. Genotypic assays detect mutations that are present as mixtures in relative proportions as low as 15–20%, even if the mutation does not affect drug susceptibility in a phenotypic assay. Moreover, genotyping can detect transitional mutations that do not cause drug resistance by themselves, but indicate the presence of selective drug pressure or the transmission of a resistant strain.

However, the interpretation of genotypic resistance tests remains challenging. Genotypic results are often interpreted by consulting "look-up tables" of drug resistance mutations. Regularly updated tables with a compilation of resistance associated mutations in protease, RT and the envelope protein gp41 can be obtained from the home page of the International AIDS Society – USA [51]. For example, within the protease more than 50 mutations at 33 different positions are listed. A more direct approach to the interpretation of genotypic data is based on rule sets encoded in computer programs that classify the virus as resistant, potentially resistant or susceptible to each drug (Section 3.2). Constructing these sets of rules is a lengthy and complicated process. Any additional information on drug resistance that is available requires frequent updating of the set of rules. Furthermore, extensive variation exists among the different available interpretation systems regarding drug activity and how drug resistance is scored [78, 88].

2.3.2 Phenotypic Resistance Testing

Similar to the procedure of genotypic assays, the first step in a phenotypic assay involves extraction of HIV-1 RNA from plasma, followed by reverse transcription and amplification of the genes of interest. These amplified genes are then used to generate recombinant viruses that can be tested for susceptibility to protease and RT inhibitors. Results of phenotypic testing are usually expressed in terms of the resistance factor (RF) which is the fold-change in susceptibility of the test sample compared with a fully susceptible control isolate.

Phenotypic testing in clinical settings is expected to be most useful for isolates with unusual combinations of resistance mutations or mutations not yet described and developing under new drugs. For example, the mutation L76V in the protease arises under several PIs in the background of other mutations that already confer high level PI resistance. Hence, current guidelines interpret such a genotype as multi-PI resistant. However, phenotypic analysis of strains harboring the L76V mutation revealed susceptibilities to saquinavir and atazanavir comparable to those of wild-type viruses, indicating that the L76V mutation has a resensitizing effect [70]. Phenotypic testing may also be useful in combination with therapeutic drug monitoring for designing salvage regimens in heavily treated patients whose viruses contain multiple drug resistance mutations.

However, the key question of whether a patient will respond to a particular drug or not remains open. Therefore, the primary challenge for interpreting phenotypic results is to predict clinical response by using data from clinical trials, studies or large cohorts in order to correlate genotypes and phenotypes (RFs) to virological response.

3 Prediction of Phenotypic Resistance from Genotypes

Phenotypic drug resistance can only be measured in a laborious virus assay (Section 2.3). On the other hand, genotyping is much more efficient, but the interpretation of the resulting sequence data is challenging. Computational methods can help interpret sequence data by predicting phenotypic traits from viral genotypes. Such genotype–phenotype relations are much easier to study if the phenotype is determined by a well-defined laboratory experiment, rather than from *in vivo* phenotypes that depend on many factors, which can confound the analysis. Therefore, predicting the *in vitro* resistance phenotype from the genotype is currently the most common approach to automatic sequence interpretation.

3.1 Drug Resistance Data

Models for resistance phenotype prediction are learned from data that have been generated in the course of routine resistance testing. Sequences are obtained from clinical samples derived from patients failing antiretroviral therapy. For each sample, the isolated virus has been genotyped (its drug targets have been sequenced) and phenotyped (tested for its replication capacity in the presence of a drug). Genotypes are represented by vectors, one entry for each position in the multiple protein sequence alignment taking one out of 21 values that represent the 20 amino acids and the gap symbol. Typically, the full protease (99 amino acids) and the 5$'$ part of the RT (amino acid positions 250–335) are sequenced. Some prediction methods require the input space to be a Euclidean vector space. In this case, each possible mutation or amino-acid change at each position is represented by an indicator variable.

Phenotypes are reported in terms of the RF, which is a single real-valued number. Coefficients of variation between 10 and 60% have been reported for this quantity [107]. On the other hand, the determination of genotypes by standard cycle sequencing is highly reproducible, but the common population sequencing strategy detects only those variants that are present in at least 15–20% of viruses in the population.

Predicting the resistance phenotype from the genotype means solving, for each drug, a regression problem. Predictors are the sequence positions of the drug target, and response is the resistance factor. Alternatively, we may consider the related binary classification problem induced by choosing a drug-specific cutoff to define a *susceptible* and a *resistant* class of viruses. Similarly multi-class classification problems arise by defining intermediate levels of resistance. Several statistical and machine learning methods have been applied to this high-dimensional, noisy data.

3.2 Methods of Phenotype Prediction

The VirtualPhenotype® (Virco, Mechelen, Belgium) is a quantitative phenotype prediction method based on a pattern search in a database of genotype–phenotype pairs [59]. From a query sequence, all mutations for a predefined set of sequence positions are extracted and all phenotypes in the database matching this pattern are retrieved. The predicted phenotype is the average resistance factor of these matches. Thus, this approach can be regarded as a nearest-neighbor method, where similarity is coached in terms of sharing a predefined set of mutations. The regression problem has also been approached by linear stepwise regression [110] and by artificial neural networks [32, 108]. For example, Wang and Larder [108] use a one-hidden layer neural network to predict resistance to lopinavir from a selection of protease sequence positions. Support vector machines (SVMs) have been applied in the regression and classification setting to all drugs [3, 11]. Since the SVM learning strategy is particularly well suited for high-dimensional data, this approach does not require the selection of a subset of mutations. The performance of linear SVMs was not improved significantly by the use of standard nonlinear kernels.

Other methods are advantageous if the goal is not only to maximize predictive power, but also to derive insight about the genotype–phenotype relation from the learned model. For example, decision tree (or recursive partitioning) methods yield models that are directly interpretable by human experts [10, 90]. Decision trees can elucidate the effect of mutational patterns on the resistance phenotype. For example, analysis of 450 genotype–phenotype pairs revealed concise models incorporating only four to seven sequence positions. Moreover, decision trees can model the effect of a mutation in the context of other mutations. In particular, some decision trees display resensitization or hypersusceptibility effects. For example, zidovudine resistance induced by mutation T215Y in the RT may be reverted by mutations L74V/I and M184I/V. The latter substitution can also resensitize tenofovir-resistant strains [112]. Likewise, the mutation N88S in the protease gene has been found to increase the susceptibility to amprenavir.

Other methods are more concerned with the statistical modeling of genotype–phenotype relations that yields procedures for hypothesis testing. Foulkes and DeGruttola [38] use cluster analysis to identify groups of similar genotypes and to define superclusters of phenotypically similar genotype groups. They introduce a statistical framework that allows for the calculation of the probability that a sequence from one supercluster is more resistant to a certain drug than a sequence from another supercluster. DiRienzo and coworkers [29] build prediction models in a forward-stepwise manner. This approach can identify specific mutational patterns that are most influential in predicting phenotype. Combinations of codons are tested for significant concordant or discordant associations with the occurrence of a mutation. Both statistical approaches have been applied to the PI indinavir.

Finally, several clinical and virologic research groups have set up scoring systems for relating sequence variations to drug resistance or the likelihood of therapy failure [80, 85, 91, 104, 106]. Based on lists of mutations that have been linked to drug resistance (Section 2) these authors provide hand-made classifiers. Prediction models are encoded as sets of rules that have been derived from the scientific literature and personal experience. These systems predict resistance to a drug in terms of one of two to five levels ranging from *susceptible* to *resistant*.

3.3 Comparisons

The performance of classifiers is usually assessed in terms of their accuracy, i.e. by the percentage of correctly classified cases in a test set. However, accuracy provides only a very limited picture of the characteristics of the trained classification model. For example, scoring classifiers are more appropriately analyzed by means of receiver operating characteristic (ROC) curves [92]. Figure 3 displays four different measures of interest for a linear SVM model that has been trained on 650 protease sequences in order to separate samples resistant to saquinavir from those susceptible to saquinavir. Figure 3(a) shows density estimates of the predicted scores for the resistant and the susceptible subgroups of viruses. Although the two classes are fairly well separated, there is some overlap. In choosing a cutoff for the prediction of resistance and susceptibility, respectively, the accuracy of the classifier will vary as shown in Figure 3(b). This dependency can be analyzed in more detail by plotting

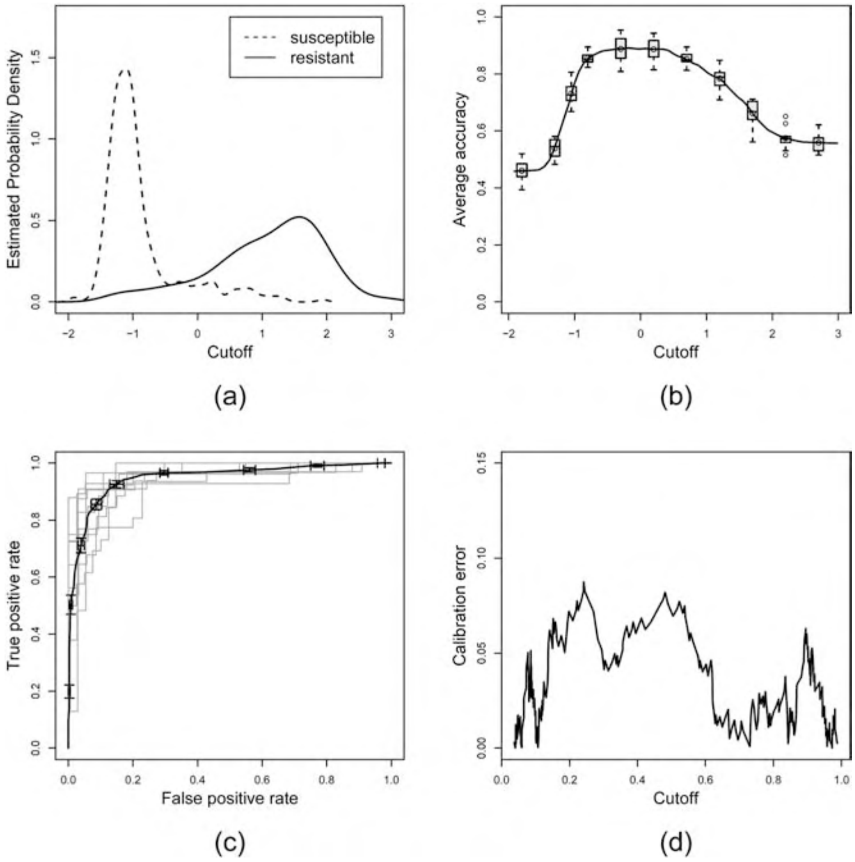


Figure 3 Performance measures for a scoring classifier that separates viruses resistant to saquinavir from those susceptible to saquinavir: density estimates of the predicted scores for the two classes (a), accuracy as a function of the cutoff used for prediction (b), ROC curve (c) and calibration error of predicting class membership probability as a function of the cutoff (d).

the false positive rate against the true positive rate. The resulting ROC curve is displayed in Figure 3(c). Finally, a scoring classifier (a method that classifies samples based on a score by choosing a cutoff) can also be used to estimate the confidence of predictions. It is important to assess whether these estimates are well calibrated. The percentage of resistance predictions at a certain confidence level should be close to that level of confidence. For example, if 80% confidence is estimated for resistance predictions at a certain cutoff, then a well-calibrated classifier would, at this cutoff, assign about 80% of the samples to the resistant class. Figure 3(d) shows the classifier’s performance in calibrating the confidence estimates [93].

Depending on the concrete application scenario, a low false-positive rate or low false-negative rate may be more desirable than minimizing the overall accuracy. For example, for heavily pretreated patients, it can be very difficult to identify active drugs for a new regimen. In this situation, a high sensitivity in detecting potentially active drugs may be required. On the other hand, for untreated patients one may want to take a more conservative approach and exclude all inactive drugs from the initial therapy with very high probability. ROC curves and related performance measures can help identifying such classifiers.

4 Development of Resistance-associated Mutations

The evolution of HIV has received considerable attention over the past 15 years not only because of its relevance for treating HIV-infected patients, but also because HIV provides an excellent model system for testing evolutionary theories. The general interest in the population genetics of HIV stems from its extraordinary evolutionary dynamics, which permit the observation of effects that occur on timescales several orders of magnitude larger in cellular organisms.

4.1 Viral Evolution

The intra-patient virus population is characterized by high rates of mutation and turnover together with high virus titers. In this situation the large majority of virions and productively infected cells turn over every day. Erroneous reverse transcription introduces substantial diversity into the population. It is estimated that any single RNA genome copied from a template molecule contains an average of 0.1–1 mutation relative to the template. The census population size (the actual number of viruses) within an untreated patient can reach 10^7–10^8 infected cells. However, the effective population size (the number of viruses in an idealized population that would show the same amount of genetic variation under random genetic drift) has been estimated to be much smaller, namely between 10^3 and 5×10^5 cells per host [16, 86]. This quantity determines the dominating mode of evolution as follows. A large effective population size implies deterministic evolution driven by mutation and selection, whereas a small effective population size implicates stochastic evolution driven by random genetic drift. The randomness in small populations is due to the finite sampling effects that arise in choosing individuals for reproduction. In very large populations even small fitness differences are recognized and the outcome of the evolutionary process is predictable. There is support for both evolutionary regimes, even in the special case of evolution under drug therapy.

Multi-nucleoside resistance, i.e. resistance to multiple NRTIs, can develop along different evolutionary pathways. This fact has been interpreted as evidence for stochastic evolution. For example, viruses in patients treated with zidovudine and didanosine follow one of two mutually exclusive pathways, apparently by chance [17]. Similarly, resistance to the PI indinavir develops by the accumulation of four to seven of approximately 12 different mutations. However, almost no two isolates from treated patients have been shown to contain the same combination of mutations. Likewise, resistance to ritonavir was observed to develop uniquely in each of five patients treated with that drug suggesting stochastic evolution of resistance patterns [74].

Apparent deterministic evolution can also be observed during the development of drug resistance. For example, single-nucleotide changes produce the mutations M184V and K103N that confer a high level of resistance to the NRTI lamivudine and the NNRTI nevirapine, respectively. These mutants are selected in over 90% of patients after a few weeks or even days of monotherapy. Convergent evolution has also been observed in protease sequences isolated from patients that were under multiple drug therapies for at least 2 years. Out of five intra-patient virus populations that had escaped selective drug pressure, identical amino acid replacements were observed in all five patients at two different sites [23].

Despite these cases in favor of deterministic evolution, selection appears to be detected less frequently than one would expect from HIV facing strong immune responses and antiviral therapy. This may indicate a more prominent role of genetic drift, but it might also be a limitation of the methodology to detect selection. The most popular test for detecting selection compares, for each site under consideration, the number synonymous mutations to the number of nonsynonymous mutations. In practice, the test can suffer from averaging the test statistic over a sequence region thereby limiting its power if selective pressure acts only on a few sites [100]. Alternative tests are based on the coalescent (a modeling framework in which two viral lineages converge in a common ancestral sequence, going backwards in time) [101] or on a Bayesian approach to sequence evolution [73].

In addition to classical phylogenetic and population genetics methods, rapidly evolving RNA populations have also been modeled as molecular quasispecies. This concept has been developed by Eigen and coworkers [34, 35] in order to describe populations of self-replicating RNA molecules as an ensemble of closely related genomes. The mutant spectrum depends on individual replication rates and hence reflects the surrounding fitness landscape. Since selection operates on this mutant cloud rather than on an individual strain, evolution is biased by the internal structure of the quasispecies. In particular, selection may favor clouds of genotypes interconnected by mutation over isolated genotypes, even if the average fitness of the cluster is lower than that of the individual. The suitability of quasispecies models for real RNA virus populations is subject to current debate [30,34,48,49,111].

Apart from random genetic drift, mutation and selection, there are additional factors that have a significant impact on evolving virus populations. For example, migration occurs within hosts via infection of multiple cells, tissues and organs. Migration also occurs between hosts as viruses spread into different human subpopulations and geographic regions. Migration results in complex spatial patterns and implies the necessity of studying structured populations [45]. Another important factor is homologous recombination, which can occur when a cell is coinfected with two different strains [53]. In HIV-1, recombination has been shown to occur approximately two to three times per genome per replication cycle. This is about 10 times as much as the mutation rate and an extremely high rate of recombination given the small genome size [52]. Recombination often biases parameter estimates of evolutionary models, but explicit modeling of recombination is notoriously difficult. Likewise, the role of recombination in the development of drug resistance is not well understood [15].

4.2 Learning Mutational Pathways

In the context of HIV drug resistance we are particularly concerned with the determinants and mode of the development of genetic changes that confer phenotypic resistance. If drug pressure is continuous and uniform, viral evolution is characterized by the accumulation of resistance-associated mutations. This accumulation occurs in a nonuniform, stochastic fashion and gives rise to coexisting evolutionary pathways. Understanding this evolutionary process is important for estimating how close a virus is to escaping from drug pressure. Mutagenetic trees, a family of probabilistic graphical models, have been developed for estimating the rate and order of occurrence of resistance-associated mutations in the viral drug targets.

Consider a set of n specific amino-acid changes (mutations) that develop under drug treatment. A mutagenetic tree for these n mutations is a connected branching on $\{0, \dots, n\}$ rooted at 0. Each vertex $v \neq 0$ represents the binary random variable X_v that indicates the occurrence of mutation v. Associating probability parameters θ_v with the tree edges one obtains a directed acyclic graphical model with transition matrices:

$$\Big(\Pr(X_v = b | X_{\mathrm{pa}(v)} = a)\Big)_{a,b=0,1} = \begin{pmatrix} 1 & 0 \\ 1-\theta_v & \theta_v \end{pmatrix},$$

where $\mathrm{pa}(v)$ denotes the parent of v in the tree. The first row of this matrix reflects the model assumptions that in the underlying evolutionary process mutations are nonreversible and that a mutation can occur only if all of its ancestor mutations have already occurred. By definition, mutagenetic trees are stochastic models, but parameter values θ_v close to 0 or 1 result in near deterministic behavior.

A mutagenetic tree defines a probability distribution on the set of all possible mutational patterns. In particular, this model family includes linear-path models (chains) and the model of complete independence given by the star topology. The complete family of mutagenetic tree models can be characterized by their algebraic invariants [6]. The key observation is that the binary state vectors that are compatible with the model, i.e. the mutational patterns which have positive probabilities, form a finite distributive lattice. This combinatorial structure allows for a simple enumeration scheme of (a basis of) all invariants.

Mutagenetic trees can be reconstructed from data as maximum-weight branchings of the complete graph on $n + 1$ vertices. This combinatorial optimization problem can be solved efficiently by Edmonds' maximum-weight branching algorithm. The weight functional used in this algorithm involves only pairwise probabilities, which can be estimated from cross-sectional data. This method has been shown to be consistent in the sense that if the data comes from a mutagenetic tree model, the procedure is guaranteed to recover the true tree [27].

The single-tree model has been extended to mixture models of mutagenetic trees that combine several weighted trees [8]. The first tree component is a star with uniform probabilities that models the spontaneous and independent occurrence of mutations. All other components represent dependencies between mutations and are estimated from the data. The mixture model is learned by an expectation maximization (EM) algorithm that iteratively estimates the expected values of the missing data (i.e. the association of samples to the trees) and the structure and parameters of the trees. For model selection (choosing the number of tree components) either cross-validation or a modified Bayesian Information Criterion can be used that is based on an estimate of the structural redundancy between tree components [113]. Mtreemix, a software package for statistical inference with mutagenetic trees and mixtures of these, is available on the internet and described in [9].

4.3 Genetic Barrier

In order to estimate the rate of development of mutations, we assume independent Poisson processes for the occurrence of mutations with parameters λ_v. If the observed sampling time (i.e. the time on therapy) is also modeled

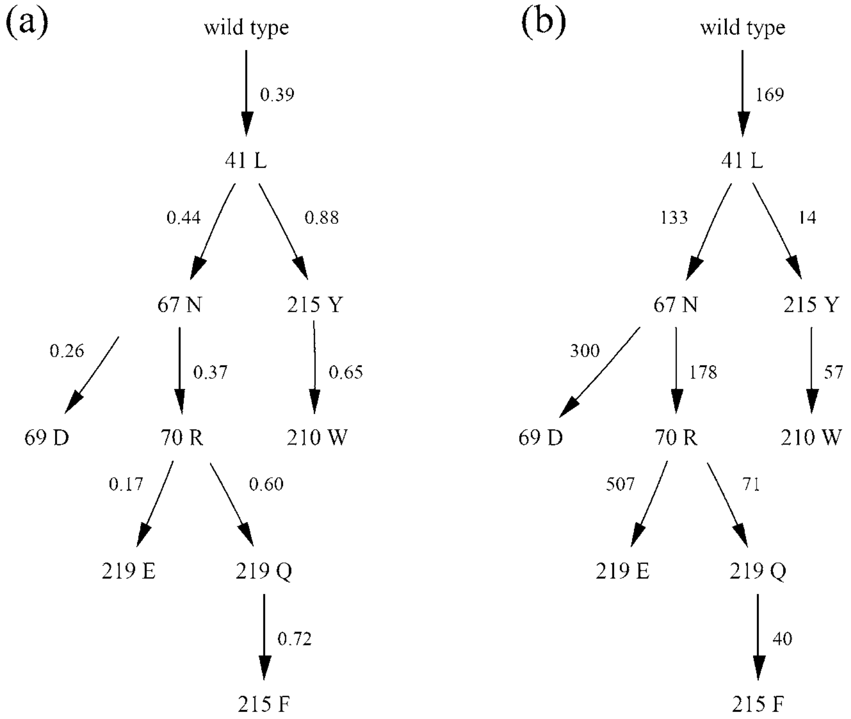


Figure 4 Mutagenetic tree for the development of resistance to zidovudine. Vertices denote amino acid changes from the wild-type; edges are labeled with conditional probabilities (a) and expected waiting times in weeks (b), respectively.

exponentially with rate λ_S, then:

$$\theta_v = \frac{\lambda_v}{\lambda_v + \lambda_S} .$$

This relation allows for the translation of the estimated conditional probabilities between mutations into the expected waiting time for the mutation to occur. Furthermore, the probabilities of the occurrence of any mutational pattern can be computed for any fixed mean waiting time. Hence, using these timed mutagenetic trees we can compare models that have initially been estimated from data sets sampled after different mean waiting times [4]. Figure 4 shows a mutagenetic tree and the corresponding timed mutagenetic tree for the development of drug resistance in the HIV RT under therapy with zidovudine. The model displays two characteristic pathways, namely the 70-219 and the 215-41 pathway [14].

The genetic barrier summarizes how difficult it is for a virus population to escape from drug pressure by developing mutations. This quantity can be estimated by means of mutagenetic trees and the phenotype predictions introduced in Section 3 [4]. Suppose we have estimated a mutagenetic tree model for the development of resistance to a certain drug. In particular, this model can be used to compute transition probabilities between mutational patterns. Using a classifier restricted to the set of n mutations we predict each mutational pattern to be either *susceptible* or *resistant*. The genetic barrier is defined as the probability of not reaching any resistant state after a fixed time period under therapy. This quantity can be calculated as the sum of the probabilities of all mutational patterns predicted as susceptible. Thus, a higher genetic barrier indicates that the virus is less likely to become resistant. In fact, a low genetic barrier has been proposed as the cause of the early virological failure, which has been observed with some combinations of otherwise very potent antiretroviral drugs [41].

For example, the genetic barrier to resistance to the nucleoside RT inhibitors zidovudine, lamivudine, and didanosine of the wild type virus has been computed under three different regimens: zidovudine mono-therapy (ZDV), double therapy with zidovudine plus lamivudine (ZDV + 3TC), and double therapy with zidovudine plus didanosine (ZDV + ddI). The genetic barriers to zidovudine resistance are ordered as ZDV + 3TC $>$ ZDV $>$ ZDV + ddI for the respective regimens. For lamivudine resistance we find ZDV $>$ ZDV + ddI $\gg$ ZDV + 3TC, and for low level didanosine resistance ZDV $>$ ZDV + ddI $>$ ZDV + 3TC. Thus, unexpectedly, zidovudine resistance appears to develop faster upon adding didanosine and a combination containing didanosine maintains a higher genetic barrier to didanosine resistance than a regimen sparing didanosine. These findings have been shown to be consistent with and to partly explain clinical data from a multi-cohort study assessing the risk of virological failure under these regimens [4, 22].

4.4 Transitions between Sequence Clusters

Foulkes and DeGruttola [39] have modeled the evolution of drug resistance as transitions between discrete states that are defined as clusters of sequences with similar patterns of mutations, rather than individual mutational patterns as used in the mutagenetic tree models. K-means clustering was applied to identify these states. A Markov model was used to estimate transition rates between states. One approach treats the state at a given time point as known, whereas another approach treats this as a latent variable. The second approach allows for consideration of the effect of minority species on the evolution of the viral populations. Both methods have been applied to protease sequences of HIVs cloned from the plasma of 170 patients who participated in a clinical study of efavirenz combination therapy. Multiple viral clones were available from each plasma sample at each measurement time point. In general, sequences of state membership that involved staying in the same state over time had the highest *a posteriori* probabilities.

NNRTI mutation data from the same clinical study was analyzed with an extended mutagenetic tree model that accounts for longitudinal and clonal samples [5]. In this mutagenetic tree hidden Markov model (HMM), progression of the virus population in time is modeled as a sequence of unobservable states, which are inferred from observed clonal samples. The hidden states, which are compatible with the given tree topology, represent the state of the population. Clonal variation is modeled by a simple error process that allows for false positives and false negatives, i.e. for observed mutations despite a wild-type population state and for wild-type residues in the presence of a mutant population state, respectively. Analysis of the NNRTI data has shown that the rates of occurrence of mutations, λ_v, can be estimated much more precisely from longitudinal than from cross-sectional data.

5 Selecting Optimal Combination Therapies

Even in the era of HAART, treatment failure is not uncommon and clinicians are frequently faced with the problem of selecting a new potent drug combination after failure of the current regimen. Preexisting drug resistance mutations can be the cause of therapy failure. In any case, viral replication under suboptimal therapy leads to the emergence of drug resistant variants (Section 2). The accumulation of resistance-associated mutations limits remaining treatment options which are available. Moreover, because of broad cross-resistance within drug classes, treatment changes cannot be based on the assumption that the virus will remain susceptible to the unused drugs. Even for therapy-naive patients not all drugs can be assumed to be active because of increasing rates of transmission of resistant viruses. The viral genotype may provide more information on the expected outcome of the new regimen than the past treatment history alone. In the following we discuss genotype-based methods for the selection of drug combinations that are estimated to maximize clinical response.

The computational task of identifying optimal antiretroviral drug combinations with respect to a given viral genotype is a typical bioinformatics problem (such as sequence alignment) in the sense that the objective function of the optimization problem is not known. In fact, we need to know the *in vivo* effect of any drug combination on any mutational pattern in order to find the best regimen. Typical clinical parameters of interest are the virus load and the number of T4 cells (Section 2). Estimating these response functions is much more challenging than actually selecting the optimal drug therapy. Indeed, the number of drug combinations is only on the order of thousands, and hence they can be enumerated. By contrast, HIV's high genetic diversity induces a much higher number of mutational patterns. Furthermore, clinical response is influenced by several factors in addition to resistance, including patient adherence, immunological status, and baseline virus load.

5.1 Clinical Databases

One way to estimate the activity of a therapeutic regimen against a viral strain is to learn this effect from observational clinical databases. This is straightforward, if we fix a combination therapy or a narrowly defined type of therapy. In this case, the same machine learning methods presented in Section 3 can be used to predict virological response. For example, King and coworkers [57] use decision trees in order to predict response to lopinavir–ritonavir-based regimens. However, if the drug combination is not fixed, direct learning from observed cohort data is limited by the amount of data necessary to derive useful models, because here the complexity of the problem depends on both mutational patterns and drug combinations. For example, Wang and coworkers use a neural network model in this situation. They estimate an approximate required database size of more than 9500 patient samples to predict virological response, if 12 drugs and 49 mutations are assumed [28, 109].

Furthermore, the distribution of drug combinations in clinical databases is heavily skewed. It reflects drug approval times and preferred treatment strategies over time rather than the full variety of potential combination therapies [2]. Thus, training on such data sets is likely to result in models that capture the features of only a few frequently observed combinations. Therefore, those datasets are not appropriate for exploring the product space of all mutational patterns and drug combinations.

5.2 Simple Scoring Functions

An alternative approach to general response prediction is to score drug combinations on the basis of single-drug effects. This implies the assumption of a model of how the effect of a drug combination depends on the single-drug effects. For example, we may use a classifier for resistance phenotype prediction and count the number of *active* drugs in a combination, i.e. the number of drugs for which the virus is predicted susceptible. This model has been used in a retrospective analysis of 332 therapy changes accompanied by genotypic resistance tests [24]. SVM-based phenotype predictions were used to define one group of patients with two or fewer active drugs and a second group of patients with three or more active drugs. Using a Cox proportional hazards model it was demonstrated that patients in the group with at most two active drugs were at significantly higher risk of virological failure (defined as two consecutive virus load measurements of more than 500 copies mL^{-1} after 24 weeks of therapy) [2].

Counting the number of active drugs in a combination therapy entails two problems. First, we have to choose a cutoff value for each drug. Second, we do not account for intermediate levels of resistance. The inherent difficulty in selecting appropriate cutoffs and the apparent loss of information can be avoided by considering the real-valued fold-change in susceptibility. However, the dynamic range of resistance factors varies by as much as two orders of magnitude between different drugs. Thus, resistance factors and their predictions are not comparable between different antiviral agents.

In order to overcome this limitation different normalizations have been proposed. In one approach, the degree of resistance is quantified relative to the biological variation of fold-change values observed among untreated patients. This distribution is normal with individual parameters for each drug. Hence, the *z*-score of a resistance factor is the number of standard deviations it differs from the mean in therapy-naive patients. Another normalization is based on phenotype predictions from genotypes obtained from both untreated and treated patients. The resulting distributions exhibit large differences in location and deviation of modes for the different drugs, but also reveal a common bimodality. Thus, we model this density by a Gaussian mixture model:

$$\lambda \times N(\mu_1 \sigma_1) + (1 - \lambda) \times N(\mu_2, \sigma_2)\,, \quad \lambda \in [0, 1]\,,$$

and estimate its parameters by the EM algorithm. This two-state model provides a data-derived definition of *susceptible* and *resistant*. By linearizing the log-likelihood ratio between these two classes, we obtain the *activity score*, which approximates the conditional probability of membership in the susceptible class given the viral genotype [7]. Thus, the activity score provides a normalized and comparable measure of resistance and it can be extended to multi-drug therapies by summing over all drugs in the combination.

5.3 Look-ahead Techniques

The genetic barrier of the virus to resistance to each of the compounds of the regimen also provides a normalized score. The genetic barrier estimates the likelihood of viral escape rather than the current level of resistance. Summing these values provides an estimate of how easy it is for the virus to escape from the selective pressure of the combination therapy. As shown in Section 4.3, this *genetic barrier score* is generally different from the genetic barrier of the drug combination, but the genetic barrier cannot be obtained for all drug combinations as it needs to be estimated from many samples derived from patients under the same regimen.

Despite these simplifications, both the activity score and the genetic barrier score have been shown to be predictive of virological response [2, 12]. In particular, classifiers based on these scores can learn concepts that are specific for the combined effect of drug combination and mutational pattern. Thus, this approach is promising for identifying individually optimized drug combinations.

In a related approach the likelihood of the virus to escape from selective pressure of a drug combination is estimated more conservatively. Applying a heuristic greedy search, the mutational neighborhood of the viral strain under consideration is explored. Successively, point mutations are introduced, the activities of the regimen against the resulting *in silico* mutants are estimated, and those variants are kept that reduce the activity of the regimen most. This strategy can be implemented efficiently by organizing the mutated sequences in a data structure called a priority queue that orders the sequences by activity. If the activity is estimated by a linear model, the new activity for a one-point mutant can be obtained by a cheap update of the previous activity avoiding a full computation of the linear function for each visited mutant. The estimated "worst case" activities at each level of depth of the sequence space search were used as inputs to a linear model and yielded significant predictions of virological success (defined as a drop in viral load of at least 2 $\log_{10}$ copies mL^{-1} after 3 months of therapy) [7].

5.4 Rules-based Approaches

As mentioned in Section 3.2 several rule-based systems aim to predict clinical response from genotypes. Based on these rule sets two computational approaches have been developed. The CTSHIV (Customized Treatment Strategy for HIV) system is a rule-based expert system designed for finding optimal resistance-avoiding combination therapies [60]. The system operates on a set of resistance-inferring rules which are applied to a patient's viral strains and nearby mutants. Drug combinations are scored by identifying the most resistant mutant and the least-resisted drug for this mutant in the respective drug combination. Nearby mutants are generated by a backward-chaining search. The optimal solution is computed by a branch and bound algorithm.

Another approach applies fuzzy logic methods to a set of expert rules [25]. Rule weights are learned from observed clinical outcomes. The resulting system has been shown to have improved results over the set of rules alone.

6 Host Genetic Profiles and Viral Evolution

6.1 Immunobiological Background

Host genetic profiles play an important role in the course and outcome of viral diseases such as HIV-1. These genetic profiles have a significant impact on the susceptibility or resistance to infection, the rate of disease progression and therefore the clinical manifestations of the disease. However, the rapid rate of viral evolution allows the virus to adapt to specific host genetic profiles. The increased understanding of the interplay between the host and the virus has provided important insights into the large variation in responses to HIV-1 infection, where some patients are extremely resistant to even becoming infected and whereas others progress to AIDS in a very short time period.

6.1.1 HLA Genes

The human major histocompatibility complex (MHC) is called the HLA system and consists of a large gene region on the short arm of chromosome 6 which contains over 100 genes [50, 97]. These genes encode proteins which are essential for adaptive and innate immunity. Of central importance in the protection against pathogens are the HLA class I genes (particularly *HLA-A*, $-B$ and *-C*) and HLA class II genes (particularly *HLA-DR*, *-DQ* and *-DP*). Class I genes encode molecules that are expressed on all nucleated cells, at varying levels and bind endogenously derived antigens and present them to T8 cells (Figure 5), leading to a cytotoxic T cell response. Class II molecules are mainly expressed on antigen-presenting cells (APCs) such as B lymphocytes, macrophages and dendritic cells. Class II molecules bind peptides of extracellular origin and present them to T4 cells, resulting in cytokine production and T cell help in antibody production. Supporting the activities of these molecules are the class III complement proteins and the inflammatory cytokine genes, which are also located in the HLA region.

HLA class I and II genes are extremely polymorphic. As of January 2005, 1179 class I alleles and 725 class II alleles had been named and the growth in the total number of observed alleles is projected to continue for some time [82]. Additionally, *HLA-B* has been confirmed as the most polymorphic gene in the human genome [71]. Allelic variation occurs both within and between different ethnic groups (http://www.allelefrequencies.net). Linkage disequilibrium (see also Chapter 38), where the alleles at one HLA locus are not randomly distributed with respect to the alleles at another HLA locus, has been described between different loci in the HLA region [68]. Due to the high number of polymorphisms in the HLA genes, most individuals are likely to be heterozygous at the most polymorphic loci. The expression of HLA alleles

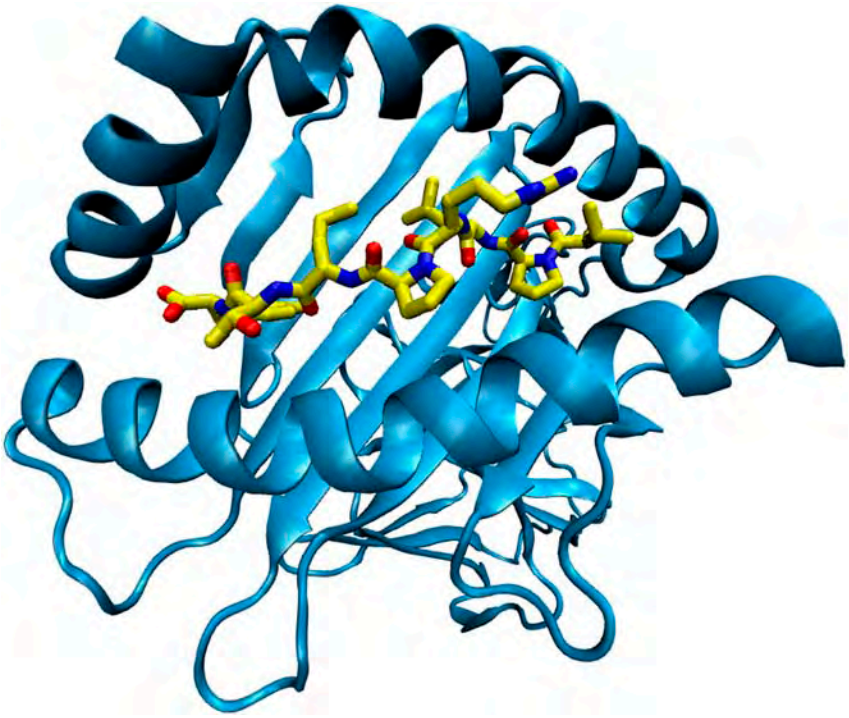


Figure 5 The crystal structure of the HLA-B*3501 allele complexed with peptide VPLRPMTY from the NEF protein (75–82) of HIV-1 [95]. The HLA molecule is shown as a blue ribbon diagram and the peptide is shown in yellow, with red oxygen atoms and blue nitrogen atoms.

is codominant, as both alleles of a locus are expressed on a cell's surface and each allele is able to present peptides to T cells.

Polymorphisms in HLA molecules are primarily concentrated among amino acids which are responsible for the binding of the foreign peptide. These amino acids are located on the floor or on the inner walls of the peptide-binding cleft. The polymorphisms cause the clefts to have different size and chemical characteristics in different allele variants. Therefore, although all HLA molecules can bind large and diverse sets of peptides, different HLA molecules have preferences in their binding affinities and specificities. Groups of *HLA-A*, -*B* and -*DR* alleles, called supertypes, have been identified which share specific binding preferences for peptides or supermotifs of a similar size, charge and amino acid composition [66,89].

Numerous studies have identified a role for HLA genotypes in HIV-1 progression and these will be discussed in detail below.

6.1.2 Chemokine Receptors

Many inflammatory and immunoregulatory cells are strongly influenced by the interaction between secreted chemokines and the chemokine receptors which are expressed on their cell surfaces. Particular attention has been focused on these receptors and their ligands because of their important role in general immunity and in the process by which HIV-1 penetrates into cells.

Two receptor–ligand families have been shown to have prominent roles in the HIV-1 infection process: the CCR and CXCR families. Two receptors which are members of the CCR family are CCR2 and CCR5. CCR5 binds the CC-motif ligands MIP-1α, LD78b (or CCL3L1), MIP-1β, RANTES, MCP-2 and HCC-1. A member of the CXCR family, CXCR4, binds SDF-1 which is encoded by the *CXCL12* gene [55].

Viruses that exclusively use the CCR5 coreceptor for cell entry are known as R5 strains and those that use CXCR4 are known as X4 strains. The binding to CCR5 typically predominates in initial infection. As the infection progresses, the virus begins to use CXCR4 instead of or in addition to CCR5. As CXCR4 is present on many more T4 cells than CCR5, this switch in coreceptor usage enables the virus to infect a far greater number of T4 cells and is associated with accelerated disease progression.

Genetic variations in the third variable loop (V3 loop) of the viral envelope protein gp120 have been associated with the switch in coreceptor usage from CCR5 to CXCR4. Machine learning methods similar to those described in Section 3.2 have been applied to predicting coreceptor usage from V3 sequence data [92]. Six statistical learning methods operating on the entire V3 loop were evaluated using cross-validation. Classifiers based on SVMs showed significantly higher area under the receiver operating characteristic curve (AUC) than other methods, with the exception of position-specific scoring matrices (PSSMs), for which the difference did not reach significance. At varying specificities, which were controlled by choosing appropriate cutoffs, SVMs dominated all other methods in terms of sensitivity. Predictions of coreceptor usage on a large longitudinal dataset agreed well with published data and showed smooth score trajectories, indicating applicability of scoring classifiers to monitoring the accumulation of coreceptor-associated sequence alterations. These models allow for careful monitoring of coreceptor usage, which is an important prerequisite for the use of coreceptor inhibitors.

Individuals who are homozygous for a 32-bp deletion in CCR5 (*CCR5-Δ32*) are almost completely protected from HIV-1 infection because of the inability of the virus to bind to the coreceptor. Rare exceptions occur when homozygotes are infected with X4 viruses that do not require CCR5 for penetration. Heterozygotes are as likely to be infected, but demonstrate a slower rate of disease progression. A number of polymorphisms in the CCR5 promoter in various ethnic groups have also been associated with increased likelihood of infection and more rapid disease progression.

In CCR2, a valine to isoleucine polymorphism (*CCR2-V64I*) has been associated with a favorable prognosis in heterozygous HIV-1-infected individuals, although the mechanism by which this occurs is still ambiguous. Studies into effects of the 3′A variant at position 801 in the untranslated region of the *CXCL12* gene which encodes SDF-1 (*SDF-1-3′A*) on HIV-1 disease progression have been less consistent. CCL3L1 is one of the ligands of CCR5. The copy number of the gene shows interindividual and interpopulation differences and influences the level of the chemokine. Possession by an individual of a copy number which was significantly lower than average for their respective racial or ethnic background has recently been associated with enhanced HIV and AIDS susceptibility [42].

6.2 Epitope Prediction

Epitopes are defined as the parts of antigens which interact with receptors of the immune system. In order to continue developing techniques which will help detect, monitor and fight diseases, it is important to know as much about the intrinsic structure of the relevant epitopes as possible. Due to the large number of epitopes and HLA molecules, prediction methods are an important tool supporting the generation of new insights into this complex process.

6.2.1 Problem Definition

Peptides which are presented by HLA class I molecules are selected in a multistep process consisting of (i) the cleavage of proteins in the cytosol into peptides by the proteosome, (ii) the N-terminal trimming in the cytosol or later in the endoplasmic reticulum (ER) by aminopeptidases, (iii) the transport of peptides into the ER by TAP proteins and (iv) the loading of the peptides onto class I molecules. These peptide–HLA complexes transit through the constitutive secretory pathway, in which transport vesicles move from the *trans*-Golgi network to the plasma membrane [83].

The peptides bound by class I molecules are generally 8–11 amino acids in length. Binding is stabilized by contacts between atoms in the free N- and C-termini of the peptide and conserved sites found at each end of the cleft in class I molecules. These interactions limit the length of the peptide accepted for binding and, therefore, the binding cleft is often described as being closed. Variations in length of the peptide seem to be accommodated by kinking of the peptide backbone.

By contrast, a number of different pathways have been describe by which HLA class II molecules become loaded with peptides. Typically, exogenous proteins are internalized by the cell or endogenous proteins resident in the endosomal system. Alternatively, antigens presumably excluded from the endosomal system, such as proteins located in the cytosol or nucleus, can be presented by class II molecules [103]. The conventional processing pathway features enzymatic unfolding, fragmentation and loading of internalized exogenous antigens within the endocytic compartment. An alternative pathway includes processing that is proteasome and TAP-dependent [102].

HLA class II molecules bind peptides that are more variable in length, approximately 11–17 amino acids long and possibly much longer. The ends of the peptide are not bound and the peptide lies in the binding groove in an extended conformation. It is held in position by its side chains, which protrude into pockets in the class II molecule's binding groove that are lined by polymorphic residues. Additionally, the peptide is stabilized by interactions between the peptide backbone and conserved amino acids of the binding groove. As HLA class II molecules are more permissive in their peptide binding than class I molecules, it is more difficult to predict which peptides will bind to particular class II molecules.

Recent research appears to suggest that there is a lack of absolute topological restrictions on HLA class I and II molecules, implying that, in principle, any protein can be presented by either molecule class. Therefore, the differences between class I and II could possibly be distinguished mainly on the basis of their binding characteristics and physiological roles [103].

6.2.2 **Methods**

The ability of predicting which regions of a target protein make good epitopes is important for the understanding of immunological processes and vaccine design. A broad variety of prediction tools have been developed in recent years, some of which deal with particular parts of the pathways, by which peptides are loaded onto HLA molecules, such as TAP transport, or alternatively with whole pathways.

Several methods have been developed for predicting proteasome cleavage. In a recent comparison it was found that the best method, NetChop, captures roughly 70% of the C-termini correctly [87]. NetChop uses neural networks trained on *in vitro* data in addition to class I ligand data.

Two approaches have been used to predict selective TAP transport. Cascaded SVMs, using two layers of SVMs, were designed to include features of the sequence and of amino acids [13]. Alternatively, a method was developed using an additive scoring function to analyze and extend the TAP-binding motif [31]. The method assumes that each amino acid makes an additive and constant contribution to the biological activity regardless of amino acid variation in the rest of the peptide. Possible interactions between amino acids are accounted for by cross-terms. This approach is also called two-dimensional quantitative structure–activity relationships (2D-QSARs) and it relies on solving the linear regression problem by using partial least squares (PLS). PLS forms new variables, named principal components, as linear combinations of the initial variables and then uses them as predictors of the dependent variable which is the TAP binding affinity.

Due to the closed nature of the binding cleft in HLA class I molecules and the well-defined epitope length, many more methods have been developed for predicting class I epitopes than class II epitopes. A comparison of some of the methods that have been used in class I epitope prediction has been made [114]. The compared methods were based on binding motifs, binding matrices, hidden Markov models (HMMs) and artificial neural networks (ANNs). The selection of the optimal method depended on the amount of available data, the biases of the data set and the intended purpose of the prediction. For datasets with more than 100 known binding peptides, HMMs and ANNs were found to be the methods of choice.

A relatively small number of prediction tools have recently been developed for class II epitope prediction. A method involving the use of motif block alignments, generated by aligning peptides known to bind specific class II molecules, were used to generate PSSMs to which a position-based weighting method was applied [79]. A novel Gibbs motif sampler method designed for recognizing weak sequence motifs has also been developed [72]. An earlier approach used a matrix-based prediction algorithm, employing an amino acid/position coefficient table deduced from the literature [94,99].

6.3 Analysis of Escape Mutations

Genetic resistance is an important aspect of the viral adaptation to the host. Its analysis is an important component of the study of HIV-1 disease. The HLA genotype of an individual appears to play an important role in the progression of the disease.

A number of large, population-based HLA class I association studies (see also Chapter 38) have been conducted [96]. The studies were done in large Caucasian and African cohorts with subtype B and C infections. They have associated *HLA-B*57* and alleles of the *HLA-B*58* supertype with low viremia and a delayed onset of AIDS. *HLA-B*27* has been consistently associated with slow progression in non-African populations. The alleles that are part of the *HLA-B*7* supertype, including *HLA-B*35*, have been associated with high viremia and fast progression.

When regions of HIV viral proteins are presented as peptides, or epitopes, by HLA molecules, an immune response may be triggered. Therefore, the less likely it is for a particular peptide to be presented by an HLA molecule, the lower the overall immune response of the patient against that particular peptide will be. In certain HLA genotypes, a mutation away from the HIV consensus sequence might be beneficial in promoting immune escape. The ability of the virus to mutate rapidly [77] enables it not only to develop resistance mutations to antiretroviral therapy, but also to adapt to the HLA genotype of the infected individual. This type of mutation is often called an escape mutation.

Once an escape mutation has occurred it has three potential evolutionary fates. It may revert to the consensus or wild-type amino acid on transmission to an individual of another HLA genotype. In this case, it will remain a target for the immune system. Alternatively, the escape mutation might be stable and the epitope that contains it will no longer be a target for the immune system. Such an escape mutation will reach fixation in a population over time as all other amino acid variants at that position are eliminated leaving only the escape mutation. The original epitope therefore also no longer exists and has become extinct. Finally, the escape mutation might not revert to consensus, but could still be contained in an epitope that is presented. In this case, the frequency of mutation will equilibrate in the population and there will be no clear consensus [62].

Several studies have examined escape mutations at a population level. An analysis of the RT gene in a large HLA-diverse cohort of Australian HIV-1-infected individuals revealed that HLA-specific polymorphisms were most evident in sites of least functional or structural constraint and were often associated with particular host HLA class I alleles. Both positive and negative HLA associations were found using standard statistical methods including a preliminary power calculation and odds ratio analyses, followed by the fitting of logistic regression models [69]. The majority of the strongest associations were confirmed in a cohort of HLA-diverse European patients using similar statistical methods [84]. In addition to this, six novel associations were identified. A study of the HIV-1 Gag polyprotein focused on two HLA alleles associated with long-term HIV control. Two escape mutations within a dominant HLA-B*57/HLA-B*5801-restricted epitope were present in about 80% of HLA-B*57/HLA-B*5801-positive individuals, but in no HLA-B*57/HLA-B*5801-negative individuals. Transmission of the mutant viruses into HLA-B*57/HLA-B*5801-negative recipients resulted in a reversion to the wild-type sequence of one of the escape mutations. The second mutation was maintained after transmission [63].

A recent analysis of the percentages of amino acid positions, at which polymorphisms away from population consensus were significantly associated with the HLA-A, -B and -C allelic groups, found that there is great variability across the HIV-1 genome. The use of McNemar tests showed that some genes have no statistically significant associations, whereas others have over 2% of their amino acids involved in significant associations [56]. Other work has shown that defined T8 cell epitopes tend to cluster in regions with distinct characteristics: conserved regions appear to have more epitopes, highly variable regions that lack epitopes bear cumulative evidence of past immune escape that may make them relatively refractive to T8 cells, and epitopes are more highly concentrated in alpha-helical regions of HIV-1 proteins [115].

A comparison of the relative contributions of HLA-A and -B alleles to the T8 cell-mediated immune response was performed using a large cohort of treatment-naive infected individuals from southern Africa. A panel of 410 overlapping synthetic peptides, spanning the entire expressed HIV genome, was generated and used to characterize the T cell responses to these peptides in interferon-α enzyme-linked immunospot (ELISPOT) assays [54]. There were marked differences in the frequency of targeting of individual peptides: some were targeted by no individuals, others where targeted by more than 25% of the cohort. Of the 30 most highly targeted peptides, 67% were HLA-B-restricted. The authors concluded that the principal focus of HIV-specific activity is at the HLA-B locus. The HLA-B gene frequencies in the population are those most likely to be most influenced by HIV disease, which is consistent with the observation that B alleles evolve more rapidly than A alleles [56]. The genetic epidemiological associations between HIV and HLA class II loci have not been as strong as those described for class I loci [18].

The observation of the influence of immune pressure on HIV-1 epitopes during HAART, showed that T8-mediated immune pressure can continue to effect viral evolution after the initiation of drug therapy [19]. Five treatment-naive patients, who achieved sizeable reductions in viral load upon initiation of HAART, were followed for 20 weeks. Each patient's response to 95 screening epitopes was evaluated prior to beginning HAART, with each of the five subjects responding to between one and three epitopes. The total number of optimized epitopes was ten and no epitope was located at a position where known resistance mutations occur. Two of the five patients displayed evidence of T8 cell-dependent antiviral pressure. In one patient, the epitope changed from an apparent escape variant prior to the initiation of therapy, to the sequence that is best recognized by the T8 response after the initiation of therapy and then to a new escape variant during continued therapy. In another study [37], a long-term-nonprogressing HLA-B*27 child managed to maintain viral loads of less than 400 copies mL^{-1} for almost a decade under a dual-nucleoside therapy until an escape mutation emerged with an immunodominant B*27-restricted CTL epitope. Subsequently, the child experienced a re-emergence of HIV-1 viremia accompanied by marked number of the CTL epitopes targeted.

Future treatment strategies for both treatment-naive and pretreated patients should not only involve the analysis of resistance mutations in order to help define treatment options, but to also take into account the influence of escape mutations since they can also play a causal role in the loss of immune control.

It may also be possible that the HLA-type of an individual has an impact on the choice of a particular mutational pathway, i.e. the sequence and type of resistance mutations a patient develops.

7 Conclusions

Computational methods for analyzing HIV drug resistance are likely to gain further importance in the future. The huge number of mutational patterns, on the one hand, and the large number of drug combinations, on the other hand, call for computational support in data management, analysis and clinical decision making. Moreover, therapeutic success appears to depend on viral genotype, therapy and host HLA type in a complicated manner that is unlikely to be captured by simple hand-made rules. Thus, the careful construction, validation and application of statistical models of therapy outcome may provide the basis for medical intervention planning. With an increasing number of drugs, optimization methods for finding optimal drug combinations will also play an increasingly important role. Hence, we expect sound statistical modeling and efficient optimization to be the key to individualized antiretroviral therapies in the future.

8 Web resources

8.1 Los Alamos HIV Databases (http://www.hiv.lanl.gov)

The Los Alamos National Laboratories maintain four different databases related to HIV infection [46]. The Sequence Database contains HIV and SIV sequences partly annotated with information on patient, sample, phenotypes and experimental techniques. The Resistance Database is a compilation of all mutations in HIV genes known to confer resistance to anti-HIV drugs. The Immunology Database offers a comprehensive, annotated listing of defined HIV epitopes. Finally, the Vaccine Trials Database provides a complete overview of HIV and SIV vaccine trials and their outcomes.

8.2 Stanford HIV Drug Resistance Database (http://hivdb.stanford.edu)

The Stanford HIV Drug Resistance Database is a curated database of sequences coding for the molecular targets of anti-HIV therapy [81]. It includes drug susceptibility data and therapy histories where publicly available. The database has been designed for the study of evolutionary and drug-related variation in the genome of HIV. The public website comprises query forms and a tool for the interpretation of genotypic resistance tests, based on various sets of expert rules.

8.3 Geno2pheno (http://www.geno2pheno.org)

Geno2pheno is a web service that provides two types of phenotype predictions from HIV DNA sequences. Geno2pheno[resistance] predicts phenotypic drug resistance to all approved antiretroviral agents using different classification and regression methods. The output also comprises normalized resistance scores (see Section 6.2) [3]. Geno2pheno[coreceptor] operates on the V3 loop of the envelope protein gp120 and predicts which coreceptor the virus can use to enter target cells [92].

8.4 IMGT/HLA Databases (http://www.ebi.ac.uk/imgt/hla)

The IMGT/HLA database is the official database of the WHO Nomenclature Committee for Factors of the HLA System [82]. It acts as a central repository for HLA gene and allele sequences. It has also recently incorporated data collected for the dictionary of HLA alleles and their serological equivalents. The database provides the basic tools needed to retrieve allele information and to perform sequence alignments. The database continues to grow with approximately between 150 and 200 sequences being added annually.

Acknowledgments

We are grateful to Tobias Sing for preparing Figure 3. N. B. acknowledges funding from the Deutsche Forschungsgemeinschaft under grant BE 3217/1-1.

References

1 Barre-Sinoussi, F., J. C. Chermann, F. Rey, et al. 1983. Isolation of a T-lymphotropic retrovirus from a patient at risk for acquired immune deficiency syndrome (AIDS). Science **220**: 868–71.

2 Beerenwinkel, N. 2004. *Computational Analysis of HIV Drug Resistance Data*. Shaker, Aachen.

3 Beerenwinkel, N., M. Däumer, M. Oette, et al. 2003. Geno2pheno: estimating phenotypic drug resistance from HIV-1 genotypes. Nucleic Acids Res. **31**: 3850–5.

4 Beerenwinkel, N., M. Däumer, T. Sing, J. Rahnenführer, T. Lengauer, J. Selbig, D. Hoffmann and R. Kaiser. 2005. Estimating HIV evolutionary pathways and the genetic barrier to drug resistance. J. Infect. Dis. **191**: 1953–60.

5 Beerenwinkel, N. and M. Drton. A mutagenetic tree hidden Markov model

for longitudinal clonal HIV sequence data. Biostatistics 2006, Mar 28, Epub ahead of print.

6 Beerenwinkel, N. and M. Drton. 2005. Mutagenetic tree models. In Pachter, L. and B. Sturmfels (eds.), *Algebraic Statistics for Computational Biology*. Oxford University Press, Oxford: 278–90.

7 Beerenwinkel, N., T. Lengauer, M. Däumer, R. Kaiser, H. Walter, K. Korn, D. Hoffmann and J. Selbig. 2003. Methods for optimizing antiviral combination therapies. Bioinformatics **19 (Suppl. 1)**: i16–25.

8 Beerenwinkel, N., J. Rahnenführer, M. Däumer, D. Hoffmann, R. Kaiser, J. Selbig and T. Lengauer. 2005. Learning multiple evolutionary pathways from cross-sectional data. J. Comput. Biol. **12**: 584–98.

9 Beerenwinkel, N., J. Rahnenführer, R. Kaiser, D. Hoffmann, J. Selbig and T. Lengauer. 2005. Mtreemix: a software package for learning and using mixture models of mutagenetic trees. Bioinformatics **21**: 2106–7.

10 Beerenwinkel, N., B. Schmidt, H. Walter, R. Kaiser, T. Lengauer, D. Hoffmann, K. Korn and J. Selbig. 2002. Diversity and complexity of HIV-1 drug resistance: a bioinformatics approach to predicting phenotype from genotype. Proc. Natl. Acad. Sci. USA **99**: 8271–6.

11 Beerenwinkel, N., B. Schmidt, H. Walter, R. Kaiser, T. Lengauer, D. Hoffmann, K. Korn and J. Selbig. 2001. Geno2pheno: interpreting genotypic HIV drug resistance tests. IEEE Intell. Syst. **16**: 35–41.

12 Beerenwinkel, N., T. Sing, T. Lengauer et al. 2005. Computational methods for the design of effective therapies against drug resistant HIV strains. Bioinformatics **21**: 3943–50.

13 Bhasin, M. and G. P. Raghava. 2004. Analysis and prediction of affinity of TAP binding peptides using cascade SVM. Protein Sci. **13**: 596–607.

14 Boucher, C. A., E. O'Sullivan, J. W. Mulder, et al. 1992. Ordered appearance of zidovudine resistance mutations during treatment of 18 human immunodeficiency virus-positive subjects. J. Infect. Dis. **165**: 105–10.

15 Bretscher, M. T., C. L. Althaus, V. Müller and S. Bonhoeffer. 2004. Recombination in HIV and the evolution of drug resistance: for better or for worse? BioEssays **26**: 180–8.

16 Brown, A. J. 1997. Analysis of HIV-1 *env* gene sequences reveals evidence for a low effective number in the viral population. Proc. Natl. Acad. Sci. USA **94**: 1862–5.

17 Brown, A. J. and D. D. Richman. 1997. HIV-1: gambling on the evolution of drug resistance? Nat. Med. **3**: 268–71.

18 Carrington, M. and S. J. O'Brien. 2003. The influence of HLA genotype on AIDS. Annu. Rev. Med. **54**: 535–51.

19 Casazza, J. P., M. R. Betts, B. J. Hill, J. M. Brenchley, D. A. Price, D. C. Douek and R. A. Koup. 2005. Immunologic pressure within class I-restricted cognate human immunodeficiency virus epitopes during highly active antiretroviral therapy. J. Virol. **79**: 3653–63.

20 Cingolani, A., A. Antinori, M. G. Rizzo, et al. 2002. Usefulness of monitoring HIV drug resistance and adherence in individuals failing highly active antiretroviral therapy: a randomized study (ARGENTA). AIDS **16**: 369–79.

21 Coffin, J. M. 1995. HIV population dynamics *in vivo*: implications for genetic variation, pathogenesis, and therapy. Science **267**: 483–9.

22 Collaboration of HIV Cohorts. 2004. Nucleoside analogue use before and during highly active antiretroviral therapy and virus load rebound. J. Infect. Dis. **190**: 675–87.

23 Crandall, K. A., C. R. Kelsey, H. Imamichi, H. C. Lane and N. P. Salzman. 1999. Parallel evolution of drug resistance in HIV: failure of nonsynonymous/synonymous substitution rate ratio to detect selection. Mol. Biol. Evol. **16**: 372–82.

24 De Luca, A., A. Cingolani, S. Di Giambenedetto, et al. 2003. Variable prediction of antiretroviral

treatment outcome by different systems for interpreting genotypic human immunodeficiency virus type 1 drug resistance. J. Infect. Dis. **187**: 1934–43.

25 De Luca, A., M. Venditti, F. Baldini, et al. 2002. Construction, training and clinical validation of an inferential interpretation system for genotypic HIV-1 drug resistance based on fuzzy rules learning from virological outcomes. Antivir. Ther. **7**: 571.

26 DeGruttola, V., L. Dix, R. D'Aquila, et al. 2000. The relation between baseline HIV drug resistance and response to antiretroviral therapy: re-analysis of retrospective and prospective studies using a standardized data analysis plan. Antivir. Ther. **5**: 41–8.

27 Desper, R., F. Jiang, O. P. Kallioniemi, H. Moch, C. H. Papadimitriou and A. A. Schaffer. 1999. Inferring tree models for oncogenesis from comparative genome hybridization data. J. Comput. Biol. **6**: 37–51.

28 DiRienzo, A. G. and V. DeGruttola. 2002. Collaborative HIV resistance-response database initiatives: sample size for detection of relationships between HIV-1 genotype and HIV-1 RNA response using a non-parametric approach. Antivir. Ther. **7**: S71.

29 DiRienzo, A. G., V. DeGruttola, B. Larder and K. Hertogs. 2003. Nonparametric methods to predict HIV drug susceptibility phenotype from genotype. Stat. Med. **22**: 2785–98.

30 Domingo, E. 2002. Quasispecies theory in virology. J. Virol. **76**: 463–5.

31 Doytchinova, I., S. Hemsley and D. R. Flower. 2004. Transporter associated with antigen processing preselection of peptides binding to the MHC: a bioinformatic evaluation. J. Immunol. **173**: 6813–9.

32 Draghici, S. and R. B. Potter. 2003. Predicting HIV drug resistance with neural networks. Bioinformatics **19**: 98–107.

33 Durant, J., P. Clevenbergh, P. Halfon, et al. 1999. Drug-resistance genotyping in HIV-1 therapy: the VIRADAPT randomised controlled trial. Lancet **353**: 2195–9.

34 Eigen, M. 1993. The Fifth Paul Ehrlich Lecture. Virus strains as models of molecular evolution. Med. Res. Rev. **13**: 385–98.

35 Eigen, M., J. McCaskill and P. Schuster. 1988. Molecular quasispecies. J. Phys. Chem. **92**: 6881–91.

36 Fauci, A. S., S. M. Schnittman, G. Poli, S. Koenig and G. Pantaleo. 1991. NIH conference. Immunopathogenic mechanisms in human immunodeficiency virus (HIV) infection. Ann. Intern. Med. **114**: 678–93.

37 Feeney, M. E., Y. Tang, K. A. Roosevelt, A. J. Leslie, K. McIntosh, N. Karthas, B. D. Walker and P. J. Goulder. 2004. Immune escape precedes breakthrough human immunodeficiency virus type 1 viremia and broadening of the cytotoxic T-lymphocyte response in an HLA-B27-positive long-term-nonprogressing child. J. Virol. **78**: 8927–30.

38 Foulkes, A. S. and V. DeGruttola 2002. Characterizing the relationship between HIV-1 genotype and phenotype: prediction-based classification. Biometrics **58**: 145–56.

39 Foulkes, A. S. and V. DeGruttola. 2003. Characterizing the progression of viral mutations over time. J. Am. Stat. Ass. **98**: 859–867.

40 Freed, E. O. and M. A. Martin. 2001. Human immunodeficiency viruses and their replication. In Knipe, D. M. and P. M. Howley (eds.), *Fields Virology*. Lippincott Williams & Wilkins, Philadelphia, PA: 1971–2041.

41 Gallant, J. E., A. E. Rodriguez, W. G. Weinberg, et al. 2005. Early virologic nonresponse to tenofovir, abacavir, and lamivudine in HIV-infected antiretroviral-naive subjects. J. Infect. Dis. **192**: 1921–30.

42 Gonzalez, E., H. Kulkarni, H. Bolivar, et al. 2005. The influence of CCL3L1 gene-containing segmental duplications on HIV-1/AIDS susceptibility. Science **307**: 1434–40.

43 Gottlieb, M. S., R. Schroff, H. M. Schanker, J. D. Weisman, P. T. Fan, R. A. Wolf and A. Saxon. 1981.

Pneumocystis carinii pneumonia and mucosal candidiasis in previously healthy homosexual men: evidence of a new acquired cellular immunodeficiency. N. Engl. J. Med. **305**: 1425–31.

44 Hahn, B. H., G. M. Shaw, K. M. De Cock and P. M. Sharp. 2000. AIDS as a zoonosis: scientific and public health implications. Science **287**: 607–14.

45 Hey, J. and C. A. Machado. 2003. The study of structured populations – new hope for a difficult and divided science. Nat. Rev. Genet. **4**: 535–43.

46 *HIV Sequence Compendium*. 2000. Los Alamos National Laboratory, Los Lamos, NM.

47 Ho, D. D., T. Moudgil and M. Alam. 1989. Quantitation of human immunodeficiency virus type 1 in the blood of infected persons. N. Engl. J. Med. **321**: 1621–5.

48 Holland, J. J., J. C. De La Torre and D. A. Steinhauer. 1992. RNA virus populations as quasispecies. Curr. Top. Microbiol. Immunol. **176**: 1–20.

49 Holmes, E. C. and A. Moya. 2002. Is the quasispecies concept relevant to RNA viruses? J. Virol. **76**: 460–5.

50 Horton, R., L. Wilming, V. Rand, et al. 2004. Gene map of the extended human MHC. Nat. Rev. Genet. **5**: 889–99.

51 International AIDS Society – USA. 2006. *HIV Drug Resistance Mutations*. http://www.iasusa.org/resistance_mutations/.

52 Jetzt, A. E., H. Yu, G. J. Klarmann, Y. Ron, B. D. Preston and J. P. Dougherty. 2000. High rate of recombination throughout the human immunodeficiency virus type 1 genome. J. Virol. **74**: 1234–40.

53 Jung, A., R. Maier, J. P. Vartanian, et al. 2002. Multiply infected spleen cells in HIV patients. Nature **418**: 144.

54 Kalyuzhny, A. E. 2005. *Chemistry and Biology of the ELISPOT Assay: Handbook of ELISPOT*. R & D Systems, Minneapolis, MN: 15–31.

55 Kaslow, R. A., T. Dorak and J. J. Tang. 2005. Influence of host genetic variation on susceptibility to HIV type 1 infection. J. Infect. Dis. **191 (Suppl. 1)**: S68–77.

56 Kiepiela, P., A. J. Leslie, I. Honeyborne, et al. 2004. Dominant influence of HLA-B in mediating the potential co-evolution of HIV and HLA. Nature **432**: 769–75.

57 King, M., D. Kempf, J. Isaacson, et al. 2002. Using classification trees to explore relationships between viral genotype and response to lopinavir/ritonavir-based regimens. Antivir. Ther. **7**: S82.

58 Korber, B., M. Muldoon, J. Theiler, et al. 2000. Timing the ancestor of the HIV-1 pandemic strains. Science **288**: 1789–96.

59 Larder, B. A., S. D. Kemp and K. Hertogs. 1999. Quantitative prediction of HIV-1 phenotypic drug resistance from genotypes: the virtual phenotype (VirtualPhenotype). Antivir. Ther. **5**: 49.

60 Lathrop, R. and M. Pazzani. 1999. Combinatorial optimization in rapidly mutating drug-resistant viruses. J. Comb. Opt. **3**: 301–20.

61 Lemey, P., O. G. Pybus, B. Wang, N. K. Saksena, M. Salemi and A. M. Vandamme. 2003. Tracing the origin and history of the HIV-2 epidemic. Proc. Natl. Acad. Sci. USA **100**: 6588–92.

62 Leslie, A., D. Kavanagh, I. Honeyborne, K. Pfafferott, et al. 2005. Transmission and accumulation of CTL escape variants drive negative associations between HIV polymorphisms and HLA. J. Exp. Med. **201**: 891–902.

63 Leslie, A. J., K. J. Pfafferott, P. Chetty, et al. 2004. HIV evolution: CTL escape mutation and reversion after transmission. Nat. Med. **10**: 282–9.

64 Levin, B. R., M. Lipsitch and S. Bonhoeffer. 1999. Population biology, evolution, and infectious disease: convergence and synthesis. Science **283**: 806–9.

65 Louwagie, J., F. E. McCutchan, M. Peeters, et al. 1993. Phylogenetic analysis of gag genes from 70 international HIV-1 isolates provides evidence for multiple genotypes. AIDS **7**: 769–80.

66 Lund, O., M. Nielsen, C. Kesmir, et al. 2004. Definition of supertypes for HLA molecules using clustering of specificity matrices. Immunogenetics **55**: 797–810.

67 McMichael, A. and P. Klenerman. 2002. HIV/AIDS. HLA leaves its footprints on HIV. Science **296**: 1410–1.

68 Miretti, M. M., E. C. Walsh, X. Ke, et al. 2005. A high-resolution linkage-disequilibrium map of the human major histocompatibility complex and first generation of tag single-nucleotide polymorphisms. Am J. Hum. Genet. **76**: 634–46.

69 Moore, C. B., M. John, I. R. James, F. T. Christiansen, C. S. Witt and S. A. Mallal. 2002. Evidence of HIV-1 adaptation to HLA-restricted immune responses at a population level. Science **296**: 1439–43.

70 Müller, S. M., M. Däumer, R. Kaiser, H. Walter, R. Colonno, P. Braun and K. Korn. 2004. Susceptibility to saquinavir and atazanavir in highly protease inhibitor (PI) resistant HIV-1 is caused by lopinavir-induced drug resistance mutation L76V. Antivir. Ther. **9**: S44.

71 Mungall, A. J., S. A. Palmer, S. K. Sims, et al. 2003. The DNA sequence and analysis of human chromosome 6. Nature **425**: 805–11.

72 Nielsen, M., C. Lundegaard, P. Worning, C. S. Hvid, K. Lamberth, S. Buus, S. Brunak and O. Lund. 2004. Improved prediction of MHC class I and class II epitopes using a novel Gibbs sampling approach. Bioinformatics **20**: 1388–97.

73 Nielsen, R. and J. P. Huelsenbeck. 2002. Detecting positively selected amino acid sites using posterior predictive *p*-values. Pac. Symp. Biocomput. Lihue, Hawaii, January 3–7, 2002: 576–88.

74 Nijhuis, M., C. A. Boucher, P. Schipper, T. Leitner, R. Schuurman and J. Albert. 1998. Stochastic processes strongly influence HIV-1 evolution during suboptimal protease-inhibitor therapy. Proc. Natl. Acad. Sci. USA **95**: 14441–6.

75 Nijhuis, M., S. Deeks and C. Boucher. 2001. Implications of antiretroviral resistance on viral fitness. Curr. Opin. Infect. Dis. **14**: 23–8.

76 Pantaleo, G., C. Graziosi and A. S. Fauci. 1993. New concepts in the immunopathogenesis of human immunodeficiency virus infection. N. Engl. J. Med. **328**: 327–35.

77 Perelson, A. S., A. U. Neumann, M. Markowitz, J. M. Leonard and D. D. Ho. 1996. HIV-1 dynamics *in vivo*: virion clearance rate, infected cell life-span, and viral generation time. Science **271**: 1582–6.

78 Ravela, J., B. J. Betts, F. Brun-Vezinet, et al. 2003. HIV-1 protease and reverse transcriptase mutation patterns responsible for discordances between genotypic drug resistance interpretation algorithms. J. Acquir. Immune Defic. Syndr. **33**: 8–14.

79 Reche, P. A., J. P. Glutting, H. Zhang and E. L. Reinherz. 2004. Enhancement to the RANKPEP resource for the prediction of peptide binding to MHC molecules using profiles. Immunogenetics **56**: 405–19.

80 Reid, C., R. Bassett, S. Day, B. Larder, V. DeGruttola and D. Winslow. 2002. A dynamic rules-based interpretation system derived by an expert panel is predictive of virologic failure. Antivir. Ther. **7**: S91.

81 Rhee, S. Y., M. J. Gonzales, R. Kantor, B. J. Betts, J. Ravela and R. W. Shafer. 2003. Human immunodeficiency virus reverse transcriptase and protease sequence database. Nucleic Acids Res. **31**: 298–303.

82 Robinson, J., M. J. Waller, P. Parham, N. de Groot, R. Bontrop, L. J. Kennedy, P. Stoehr and S. G. Marsh. 2003. IMGT/HLA and IMGT/MHC: sequence databases for the study of the major histocompatibility complex. Nucleic Acids Res. **31**: 311–4.

83 Rock, K. L., I. A. York, T. Saric and A. L. Goldberg. 2002. Protein degradation and the generation of MHC class I-presented peptides. Adv. Immunol. **80**: 1–70.

84 Roomp, K., G. Ahlenstiel, N. Beerenwinkel, J. Rockstroh, M. Däumer, U. Spengler and T. Lengauer. 2005. HLA profiles predict known and novel HIV-1 escape mutations at a population level. Presented at 2nd Int.

Immunoinformatics Symp., March 7–9, Boston, MA, USA.

85 ROUSSEAU, M. N., L. VERGNE, B. MONTES, M. PEETERS, J. REYNES, E. DELAPORTE AND M. SEGONDY. 2001. Patterns of resistance mutations to antiretroviral drugs in extensively treated HIV-1-infected patients with failure of highly active antiretroviral therapy. J. Acquir. Immune Defic. Syndr. **26**: 36–43.

86 ROUZINE, I. M. AND J. M. COFFIN. 1999. Linkage disequilibrium test implies a large effective population number for HIV *in vivo*. Proc. Natl. Acad. Sci. USA **96**: 10758–63.

87 SAXOVA, P., S. BUUS, S. BRUNAK AND C. KESMIR. 2003. Predicting proteasomal cleavage sites: a comparison of available methods. Int. Immunol. **15**: 781–7.

88 SCHMIDT, B., H. WALTER, N. ZEITLER AND K. KORN. 2002. Genotypic drug resistance interpretation systems – the cutting edge of antiretroviral therapy. AIDS Rev. **4**: 148–56.

89 SETTE, A. AND J. SIDNEY. 1999. Nine major HLA class I supertypes account for the vast preponderance of HLA-A and -B polymorphism. Immunogenetics **50**: 201–12.

90 SEVIN, A. D., V. DEGRUTTOLA, M. NIJHUIS, J. M. SCHAPIRO, A. S. FOULKES, M. F. PARA AND C. A. BOUCHER. 2000. Methods for investigation of the relationship between drug-susceptibility phenotype and human immunodeficiency virus type 1 genotype with applications to AIDS clinical trials group 333. J. Infect. Dis. **182**: 59–67.

91 SHAFER, R. W., R. KANTOR AND M. J. J. GONZALES. 2000. The Genetic basis of HIV-1 resistance to reverse transcriptase and protease inhibitors. AIDS Rev. **2**: 211–28.

92 SING, T., N. BEERENWINKEL AND T. LENGAUER. 2004. Learning mixtures of localized rules by maximizing the area under the ROC curve. In Proc. 16th Eur. Conf. on Artificial Intelligence (Workshop on ROC Analysis in AI), Valencia: 89–96.

93 SING, T., O. SANDER, N. BEERENWINKEL AND T. LENGAUER. 2005. ROCR: visualizing classifier performance in R. Bioinformatics **21**: 3940–1.

94 SINGH, H. AND G. P. RAGHAVA. 2001. ProPred: prediction of HLA-DR binding sites. Bioinformatics **17**: 1236–7.

95 SMITH, K. J., S. W. REID, D. I. STUART, A. J. MCMICHAEL, E. Y. JONES AND J. I. BELL. 1996. An altered position of the alpha 2 helix of MHC class I is revealed by the crystal structure of HLA-B*3501. Immunity **4**: 203–13.

96 STEPHENS, H. A. 2005. HIV-1 diversity versus HLA class I polymorphism. Trends Immunol. **26**: 41–7.

97 STEPHENS, R., R. HORTON, S. HUMPHRAY, L. ROWEN, J. TROWSDALE AND S. BECK. 1999. Gene organisation, sequence variation and isochore structure at the centromeric boundary of the human MHC. J. Mol. Biol. **291**: 789–99.

98 STINE, G. J. 2005. *AIDS Update 2005: An Annual Overview of Acquired Immune Deficiency Syndrome*. Pearson Benjamin Cummings, San Francisco, CA: 109–37.

99 STURNIOLO, T., E. BONO, J. DING, et al. 1999. Generation of tissue-specific and promiscuous HLA ligand databases using DNA microarrays and virtual HLA class II matrices. Nat. Biotechnol. **17**: 555–61.

100 SUZUKI, Y. 2001. Virus evolution. In BALDING, D., M. BISHOP AND C. CANNINGS (eds.), *Handbook of Statistical Genetics*. Wiley, New York, NY: 377–413.

101 TAJIMA, F. 1989. Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. Genetics **123**: 585–95.

102 TEWARI, M. K., G. SINNATHAMBY, D. RAJAGOPAL AND L. C. EISENLOHR. 2005. A cytosolic pathway for MHC class II-restricted antigen processing that is proteasome and TAP dependent. Nat. Immunol. **6**: 287–94.

103 TROMBETTA, E. S. AND I. MELLMAN. 2005. Cell biology of antigen processing *in vitro* and *in vivo*. Annu. Rev. Immunol. **23**: 975–1028.

104 TURAL, C., L. RUIZ, C. HOLTZER, et al. 2002. Clinical utility of HIV-1 genotyping and expert advice: the Havana trial. AIDS **16**: 209–18.

105 UNAIDS. 2005. *AIDS Epidemic Update December 2004*. http://www.unaids.org/.

106 Van Laethem, K., A. De Luca, A. Antinori, A. Cingolani, C. F. Perna and A. M. Vandamme. 2002. A genotypic drug resistance interpretation algorithm that significantly predicts therapy response in HIV-1-infected patients. Antivir. Ther. **7**: 123–9.

107 Walter, H., B. Schmidt, K. Korn, A. M. Vandamme, T. Harrer and K. Uberla. 1999. Rapid, phenotypic HIV-1 drug sensitivity assay for protease and reverse transcriptase inhibitors. J. Clin. Virol. **13**: 71–80.

108 Wang, D. and B. Larder. 2003. Enhanced prediction of lopinavir resistance from genotype by use of artificial neural networks. J. Infect. Dis. **188**: 653–60.

109 Wang, D., B. A. Larder, A. Revell, et al. 2003. A Neural network model using clinical cohort data accurately predicts virological response and identifies regimens with increased probability of success in treatment failures. Antivir. Ther. **8**: S112.

110 Wang, K., E. Jenwitheesuk, R. Samudrala and J. E. Mittler. 2004. Simple linear model provides highly accurate genotypic predictions of HIV-1 drug resistance. Antivir. Ther. **9**: 343–52.

111 Wilke, C. O. 2005. Quasispecies theory in the context of population genetics. BMC Evol. Biol. **5**: 44.

112 Wolf, K., H. Walter, N. Beerenwinkel, et al. 2003. Tenofovir resistance and resensitization. Antimicrob. Agents Chemother. **47**: 3478–84.

113 Yin, J., N. Beerenwinkel, J. Rahnenführer and T. Lengauer. 2006. Model selection for mixtures of mutagenetic trees. Stat. Appl. Genet. Mol. Biol. **5(1)**: Article 17.

114 Yu, K., N. Petrovsky, C. Schonbach, J. Y. Koh and V. Brusic. 2002. Methods for prediction of peptide binding to MHC molecules: a comparative study. Mol. Med. **8**: 137–48.

115 Yusim, K., C. Kesmir, B. Gaschen, et al. 2002. Clustering patterns of cytotoxic T-lymphocyte epitopes in human immunodeficiency virus type 1 (HIV-1) proteins reveal imprints of immune evasion on HIV-1 global variation. J. Virol. **76**: 8757–68.

41
Analyzing the Evolution of Infectious Bacteria

Dawn Field, Edward J. Feil, Gareth Wilson, and Paul Swift

Nothing makes sense except in the light of evolution.
T. Dobzhansky, 1973

1 Introduction

The sequencing of hundreds of genomes is providing new insights into the evolutionary processes that lead to the emergence and maintenance of traits related to virulence and pathogenicity. In this chapter, we discuss how bioinformatic approaches to genome analysis can be guided by evolutionary principles and complemented by phylogenetic studies of population and species. When combined, the fields of molecular evolution and bioinformatics provide a powerful set of conceptual, analytical and practical tools for mining information from genomic sequences. Likewise, pathogens are fascinating subjects for evolutionists because of the intense selective pressures associated with host–pathogen interactions. Those interested in mechanisms of pathogenesis, epidemiology, and the development of new antimicrobials and vaccines can often benefit from an understanding of the evolution of infectious bacteria.

Bioinformatics has its roots as much in molecular evolutionary theory as it does in statistics, mathematics and computing. This is because the theory of evolution provides a unifying principle with which to gain meaning from molecular data. Here, we provide an overview of how evolutionary theory helps us to better understand a biological phenomenon of pressing relevance – *pathogenesis*. We discuss the relevance of evolution to the study of infectious bacteria by reviewing key evolutionary concepts and some of the most commonly used methods and tools in molecular evolutionary biology. First, we give an overview of molecular evolutionary theory, review the number and nature of the genomes available from infectious bacteria, and provide a practical overview of relevant software and databases. We then address four areas in which an evolutionary perspective has profoundly impacted

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

our understanding of infectious bacteria. These areas are the detection of the determinants of virulence, the determination of relationships between isolates, the ubiquity and importance of lateral gene transfer and its role in niche adaptation, and the evolution of adaptive traits as a direct result of coevolution with the host. Finally, we examine the future outlook for the burgeoning field of evolutionary pathogenomics.

1.1 Introduction to Molecular Evolutionary Theory

There are many opportunities for applying evolutionary theory directly to the bioinformatic analysis of genomes. For example, the assignment of function based on homology, perhaps the most widely used concept in bioinformatics, is based on molecular evolutionary theory. If gene A is homologous (related by evolutionary descent) to gene B, it is likely to have a similar function, since both genes evolved from a common ancestor. This process of characterizing novel sequences can also be extended to the inference of similar structural properties (homology modeling; see Chapter 10). While there are potential dangers associated with assigning function (or structure) purely on the basis of sequence similarity, the value of this approach should not be underestimated.

There are three main foci to molecular evolutionary studies: the origin of life, the evolutionary history of the subsequent expansion of life forms and the manner in which molecules evolve. Reconstructing the evolutionary history and the relationships between all extant bacteria falls into the field of phylogenetics and requires knowledge of how sequences evolve. It is beyond the scope of this chapter to review all the methods and tools currently employed in phylogenetic and population genetics studies. Rather, we cover some of the key methods and tools. For a comprehensive review of phylogenetics, the reader should refer to Chapter 4.

Molecular phylogenetic tree building has a long history and developed from methods based on morphology. Evolutionary relationships can be reconstructed using a variety of molecular methodologies that involve comparisons of 16S rDNA genes, complete proteomes, restriction length fragment patterns (RFLPs), housekeeping genes, repetitive sequences, single nucleotide polymorphisms (SNPs) and other polymorphisms (Table 1). The choice of method is determined by both the question to be addressed (i.e. epidemiological or evolutionary), as well as basic features of the bacteria in question. Currently, multilocus sequence typing (MLST) is of particular importance because of its reproducibility and the ease with which data can be compared between laboratories [17, 43]. This method is based on the DNA sequencing of six to eight housekeeping, or "core", genes. Such genes are preferred as phylogenetic markers because they are conserved across strains, have relatively uniform levels of genetic variability making it easier to pool information across loci and are under strong stabilizing selection such that the majority of polymorphisms detected are likely to have arisen as the result of neutral mutations. Each unique allele encountered in a species can be given a number thus allowing for the development of strain-specific "profiles". MLST data can be used to uniquely type each processed isolate, thus facilitating epidemiological studies and the characterization of clonal complexes [25].

There are three general principles currently employed in the development of methods for reconstructing phylogenetic trees. The first uses a distance matrix built from the differences between sequences or sets of loci to build a tree. The second uses the principle of parsimony, which assumes that the correct tree is the one that requires the smallest number of past mutation events. The third, maximum likelihood, uses an explicit model of sequence evolution to judge which tree is most likely (the tree with the largest log likelihood score) given a particular data set. Despite these differences, there is strong agreement that, regardless of method, high-quality data should always produce the same tree. With lower quality, or less phylogenetically informative, data, different methods can vary in their reliability.

Constructing a tree is only the first step in inferring evolutionary relationships. It is then useful to place some estimate of confidence on the tree (*bootstrapping*), test whether the tree meets certain assumptions of the Neutral Theory and whether it contains evidence of past recombination events. The Neutral Theory states that the majority of mutations are neutral and do not influence fitness [38]. The power of this theory comes from its ability to define "null models" based on how sequences evolve under neutral assumptions. Statistically significant deviations from the expectations associated with these null models can therefore be taken as evidence for potential "non-neutral" evolutionary forces.

The observation that neutral genetic variation accumulates at a relatively constant rate has led to the formulation of the molecular clock hypothesis. If the mutation rates of neutral characters (e.g. synonymous sites in coding regions) have been empirically determined and an independent date for the divergence of two lineages is available (in the case of eukaryotes this is usually based on fossil evidence), the molecular clock can be calibrated for a particular group of organisms. The assumption of a molecular clock allows the age of a last common ancestor to be calculated, and enables the dating of nodes within a phylogenetic tree and tests for non-neutral departures from expectations. At all times, it is important to bear in mind that all trees are hypotheses and should be treated as such.

Short-term evolutionary events occur within populations and the field of population genetics is closely allied to the field of molecular evolution. Many tools and approaches are shared across these two fields, but population ge-

Table 1 A variety of molecular markers can be used to explore the phylogenetic relationships, population structure and epidemiological patterns of infectious bacteria: overview of the molecular evolutionary studies that can be employed

Method name	Application	Reference	Conclusions of study
16S rDNA	identification of uncultured pathogenic bacteria by placing new 16S sequences into the bacterial tree	60	positive identification of the uncultured bacterial species, *T. whipplei*, that causes Whipple's disease
Comparison of whole proteomes	pairwise comparison of all predicted proteins in different species to reconstruct bacterial phylogenetic trees	58, 68	strong phylogenetic signal in resulting tree and concordance with 16S rDNA tree
Ribotyping	generation of a tree of more than 500 nontypeable *H. influenzae* to examine range of variation of potential vaccine epitopes	8	use of a phylogenic strategy for vaccine development is proposed and validated
MLST	generation of trees for housekeeping genes and examination of levels of congruence between trees to determine frequency of recombination	26	unexpectedly high levels of recombination in some species; variation in levels of recombination across species
Sequence surveys	generation of sufficient sequence data to date emergence of human pathogens and examine the nature and rate of recombination	24	last common ancestor of *H. pylori* existed 2500–11 000years ago and size of recombinant fragments is unusually small
Multilocus variable number tandem repeat analysis (MLVA)	multiplexed polymerase chain reaction analysis of repetitive regions for strain identification and epidemiological studies	37	human commerce has contributed to observed geographic dispersal of *B. anthracis*
SNPs, insertion deletions, tandem repeats	Sequencing of a second strain of *B. anthracis* to find polymorphic markers	59	rare polymorphic sites are present in this genetically monomorphic species which are suitable for tracking infectious disease outbreaks

netics, which is concerned with the changes in the frequency of alleles, is more mathematical in nature. The application of population-genetic theory in genomics is still in its early days, but this new frontier in genomics holds great promise for future discovery, especially as multiple strains from a single species are sequenced and "resequencing" of large quantities of DNA becomes possible using custom microarrays [81].

Evolutionary theory also provides a range of assumptions useful in mining biologically significant patterns from genome sequences. For example, it can be used to examine the magnitude and type of selection acting upon a gene. There are several tests for selection based on comparing the ratio of synonymous to nonsynonymous substitutions between sequences [50]. Recently, an exciting method was developed that purported to detect selection in a single genome sequence [56]. However, it appears that this method may be sensitive to underlying assumptions regarding mutational processes acting at the DNA level, thus re-emphasizing the power and importance of comparative approaches in molecular evolutionary studies [50]. Furthermore, a recent study comparing multiple genomes of several bacterial species or genera demonstrated that the proportion of nonsynonymous changes decreases over time, thus it is critical to place inferences of selection within a temporal (time-dependent) perspective [62].

1.2 The Quantity and Quality of Data Available

The genomic data currently available for the evolutionary analysis of infectious bacteria are vast and rapidly expanding. It is widely recognized that our genome collection is strongly biased toward the tiny fraction of total bacterial biodiversity that enters into relationships with human hosts and leads to disease. Of the first 220 bacterial isolates sequenced, 117 are capable of causing disease. We now have genome sequences for an array of medically important microbes including those with the potential to cause epidemics, serious outbreaks of disease on a local level, commensals (bacteria that occasionally cause disease) and opportunistic pathogens that cause disease only in compromised hosts. Access to genomic data has revolutionized our understanding of the fluid nature of bacterial genomes, emphasized the importance of the evolutionary perspective when interpreting results, and spawned the new and exciting field of phylogenomics [22].

As our taxonomic sampling of genomes becomes more representative of the biodiversity in nature, so the application of evolutionary principles to the analysis becomes more powerful. The second and fourth bacterial genomes were Mycoplasmas, and the sixth and 18th genomes were both *Helicobacter pylori* strains [7]. The potential scope for comparative genomic studies has exploded since that time. At the time of writing, of the 220 bacterial isolates sequenced, 132 are from species with a single sequenced representative, while the remaining 88 isolates come from 33 species. Among these 220, the best sampled species are all capable of causing disease. These include *Staphylococcus aureus* (6), *Streptococcus pyogenes* (5), *Chlamydophila pneumoniae*

(4), *Escherichia coli* (4) and *Bacillus anthracis* (4). There are also well-sampled groups like the Mollicutes [35] and Campylobacteria [23].

1.3 A Practical Overview of Online Resources

Combining evolutionary theory and bioinformatic approaches provides a powerful conceptual and practical toolkit for the study of infectious bacteria. Databases and software useful in the study of evolutionary pathogenomics are listed in Table 2. This list is far from exhaustive, but provides a starting point for exploring the genomes of infectious bacteria, allowing the user to examine levels of conservation between genomes, construct phylogenetic relationships between sequences, and detect the presence of virulence factors and horizontally transferred sequences.

Likewise, Table 3 lists some of the more widely used taxonomic and phylogenetic resources available. The true phylogeny of bacteria is still contentious, but there are two primary sources of taxonomic information, the printed Bergey's Manual and the electronic National Center for Biotechnology Information (NCBI) taxonomy. These formalized bacterial taxonomies are a classification which is not always a true reflection of phylogenetic relationships, but a best approximation given methods available in the past. Bergey's Manual is the most authoritative source for bacterial taxonomy. It offers what it terms a "taxonomic outline", which includes all named bacteria and is now largely based on phylogenetic information derived from molecular studies of 16S ribosomal RNA genes [53]. 16S rDNA genes are widely used to reconstruct phylogenetic relationships because ribosomal RNA genes are a universal feature of prokaryotic genomes. This outline can be downloaded as a .pdf file from the web and is found in the printed volumes of Bergey's Manual. In contrast, the NCBI Taxonomy provides an online, browsable taxonomy that contains bacteria for which there is at least one sequence in GenBank [77]. While not claiming to be an authoritative source of taxonomic information, this taxonomy is based on Bergey's Manual and the contribution of a wide range of experts. It is widely used in bioinformatics, in large part, because each taxonomic group is linked electronically to all associated data held at NCBI.

Another excellent source of information on the relationships between bacteria is the Ribosomal Database Project (RDP-II), which contains over 100 000 bacterial small-subunit rDNA gene sequences in aligned and annotated format [15]. Tools available at the RDP allow the user to browse a taxonomic (phylogenetic) hierarchy of isolates and select sequences for phylogenetic analysis or download. Alternatively, newly sequenced rDNA sequences can be uploaded into the RDP, to be aligned against existing data. There are now a variety of free (phylip, MEGA, splitstree) and commercial (PAUP,

Table 2 Databases and software resources for evolutionary pathogenomic studies

Comprehensive comparative genomic databases	
Comprehensive Microbial Resource (CMR) (http://www.tigr.org/tigr-scripts/CMR2/CMRHomePage.spl)	provides access to a wide range of information and analyses about all complete bacterial genomes
Virulogenome (http://www.vge.ac.uk/index.html)	access to complete and incomplete genomes, including Artemis applet and ACT comparisons
Databases of the genomes of infectious bacteria	
Molligen (http://cbi.labri.fr/outils/molligen)	web site dedicated to mollicute genomes which allows blast searching and whole genome alignment
Oral Pathogens database (http://www.oralgen.lanl.gov)	databases of bacterial oral pathogens
Pathema (http://www.tigr.org/pathema/index.shtml)	in-depth curated analysis of pathogen genomes
Pathogen Sequencing Unit (http://www.sanger.ac.uk/Projects/Pathogens)	sequenced genomes of organisms relevant to human and animal health with related tools
STDGen and the Oral Pathogens database (http://www.stdgen.lanl.gov)	databases of genomes responsible for sexually transmitted diseases
Multiple genome alignment tools	
A Genome Comparison Tool (ACT) (http://www.sanger.ac.uk/Software/ACT)	a DNA sequence comparison viewer
Mauve (http://gel.ahabs.wisc.edu/mauve)	multiple genome alignments in the presence of large-scale evolutionary events
Multi-LAGAN (http://lagan.stanford.edu/lagan_web/index.shtml)	one of several packages in the LAGAN tool set for multiple alignment of genomes
MultiPipMaker (http://pipmaker.bx.psu.edu/ pipmaker)	summarizes similarity between multiple sequences using percentage identity plots
Multiple Genome Aligner (MGA) (http://bibiserv.techfak.uni-bielefeld.de/mga)	computes multiple genome alignments for large, closely related DNA sequences
Phylogenomic databases	
PyPhy (http://www.cbs.dtu.dk/staff/thomas/pyphy)	large-scale reconstructions of phylogenetic relationships of complete microbial genomes
Phylogenomic Display of bacterial genes (Phydbac) (http://igs-server.cnrs-mrs.fr/phydbac)	web interactive tool that displays phylogenomic profiles of bacterial protein sequences

MacClade) software packages available for phylogenetic analysis of 16S and other sequences, and Joe Felsenstein's website provides a comprehensive list of these resources (Table 3). For more on software for phylogenetic analysis, see Chapter 4.

Increasingly, there are now tools for the analysis of MLST data [36, 70]. The MLST websites (http://www.pubmlst.org and http://www.mlst. net) are portals to these resources, and offer access to all data sets (e.g. profiles and allele sequences) and a customized computing platform that contains a range of prepackaged software for bioinformatic and phylogenetic research

Table 2 (continued)

Curated databases of virulence factors	
Virulence factor database (VFDB) (http://www.mgc.ac.cn/VFs)	curated information about the virulence factors of 16 of the best-characterized bacterial pathogens
Databases of horizontally transferred sequences	
BAE-Watch (http://www.pathogenomics.bc.ca/BAE-watch.html)	database of pathogen proteins with unusually high similarity to eukaryote proteins
GenomeAtlas (http://www.cbs.dtu.dk/services/GenomeAtlas)	genomic atlases with views of compositional biases useful in detecting regions of foreign DNA
IslandPath (http://www.pathogenomics.bc.ca/IslandPathExamples.html)	identification of horizontally transferred genes and genomics islands, including pathogenicity islands

An expanded number of resources can be found in the annual issues of *Nucleic Acids Research* dedicated to descriptions of databases (January) and web servers (http://July).

(BioLinux). There are already online MLST databases for a variety of infectious bacteria, including *Burkholderia pseudomallei* (and related species), *Campylobacter jejuni*, *Campylobacter coli*, *Enterococcus faecium*, *Haemophilus influenzae*, *H. pylori*, *Moraxella catarrhalis*, *Neisseria meningitidis*, *Streptococcus agalactiae*, *S. aureus*, *Salmonella enterica*, *Staphylococcus epidermidis*, *S. pneumonia* and *Streptococcus pyogenes.* Groups interested in generating MLST data for new species are encouraged to contact the researchers hosting these sites as most of the database components can be easily reused [12].

2 Identification and Study of Determinants of Virulence and Pathogenicity

Whole-genome sequencing reveals the entire genetic complement of an isolate, and, in the case of pathogens, may reveal key aspects about phenotypic traits associated with disease and the mechanisms by which they evolved. Evidence from genomic analysis is helping to address "big-picture" questions such as, "What makes a pathogen?", "Why does pathogenicity evolve?", "Can the ability to cause disease be lost over time?" and "Where do diseases originate and why do they appear?". A crucial step in this complex process of discovery is the use of genome sequences to identify classical virulence factors. This is largely done through homology, but can be complemented by a range of *in silico* approaches which attempt to find novel virulence factors using non-homology-based reasoning. This reasoning is primarily based on clues including evidence of surface exposure, high copy number, repetitive nature, high mutation rate or uniqueness to pathogens.

Table 3 An overview taxonomic and phylogenetic of resources for studying the evolution of infectious bacteria

Resource	Description
Taxonomy	
Bergey's Manual of Systematic Bacteriology (http://www.cme.msu.edu/bergeys)	"taxonomic outline", including all species, type strains and 16S rDNA sequence data
NCBI Taxonomy (http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html)	taxonomy for sequences in GenBank with links to related resources in NCBI databases
The Taxonomy Browser (http://www.msu.edu/%7Egarrity/taxoweb)	original data analysis tool for visualizing the taxonomic relationships among prokaryotes
16S ribosomal sequence phylogenies	
Ribosomal Database Project II (http://rdp.cme.msu.edu/html)	database of ribosomal sequences and related tools for sequence alignment and phylogenetic analysis
ARB (from Latin "arbor" meaning "tree") (http://www.arb-home.de/	community standard for managing 16S data sets and building phylogenetic trees
Sequence alignment software packages	
Clustal (http://www.ebi.ac.uk/clustalw)	web interface to widely used sequence alignment program
Tcoffee & 3Dcoffee (http://igs-server.cnrs-mrs.fr/Tcoffee/tcoffee_cgi/index.cgi)	web interface to software for alignment of sequences and structures
Phylogenetic software	
CVTree (Composition Vector Tree Method) (http://cvtree.cbi.pku.edu.cn:1977/cvtree/index.php)	uses similarities in word frequencies between proteomes to build phylogenetic trees
Horizstory and Lumbermill (http://coffee.biochem.dal.ca)	a pair of phylogenetic tools to compare multiple trees and detect lateral gene transfer events
Joe Felsenstein's web pages (http://evolution.genetics.washington.edu/phylip/software.html)	comprehensive collection of links to software for phylogenetic and population genetic analysis
MEGA (Molecular Evolutionary Genetics Analysis) (http://www.megasoftware.net)	estimates evolutionary distances, reconstructs phylogenetic trees and computes statistical quantities
MacClade (http://macclade.org)	facilitates studies of character evolution using phylogenetic trees
PAUP (Phylogenetic Analysis Using Parsimony) (http://paup.csit.fsu.edu)	the most widely used and comprehensive commercial phylogenetic software package
Phylip (Phylogeny Inference Package) (http://evolution.genetics.washington.edu/phylip.html)	widely distributed free phylogeny package which implements a variety of methods
SplitsTree (http://bibiserv.techfak.uni-bielefeld.de/splits)	test whether a data set creates a bifurcating tree, outputs relationships between sequences as networks
TreeView (http://taxonomy.zoology.gla.ac.uk/rod/treeview.html)	a program for viewing phylogenetic trees
MLST related resources	
Multilocus Sequence Typing (http://mlst.net)	MLST home page
PubMLST (http://pubmlst.org)	portal which provides access to MLST resources
START (Sequence Type Analysis and Recombinational Tests) (http://pubmlst.org/software/analysis/start)	software that performs strain-specific profiles, lineage assignment and tests for recombination/selection
eBURST (http://eburst.mlst.net)	clustering algorithm to examine patterns of descent within closely related strains

2.1 Homology-based Detection

The first aim of the sequencing of the genome of a pathogen is the identification of determinants of virulence. Determining what makes an isolate capable of causing disease is a complex process. The most direct method is to look for "classical" virulence factors. This is usually done by searching new genomes for genes with homology to known virulence factors. Approaches based on homology work because it is statistically improbable that two sequences showing a high degree of sequence similarity are not related through descent [3]. Hundreds of virulence factors belonging to known gene families have been found through this method since bacterial genome sequencing began, attesting to the power of this approach. Any new sequence can be compared against the virulence factor database (VFDB) a database of known virulence factors for 16 of the best characterized bacterial pathogens [13].

2.2 Pattern-based Detection

There are several methods of finding genes involved in virulence without recourse to known homologs. For example, knowledge of the statistical properties of DNA has proven a powerful method of detecting "foreign" DNA and this is useful because horizontally transferred sequences often include determinants of virulence. Foreign regions of DNA can be detected by their skewed G + C composition, third position G + C content, codon composition or other higher level patterns. These sequences range in size from short fragments to large regions with many genes known as pathogenicity islands. The evolutionary character and importance of pathogenicity islands has recently been reviewed [64].

The statistical properties and distributions of sequences in a genome can also be used to develop searches for specific types of sequences. For example, it is known that transmembrane regions contain more hydrophobic amino acids than would be expected by chance. Therefore, algorithms have been developed to find surface-exposed transmembrane proteins *de novo* based on this fact, e.g. in the development of vaccines [55]. (See also Chapter 9 on the topic of identifying transmembrane regions in proteins.) Other genes, like the large family of PE/PPE glycine-rich genes in *Mycobacterium tuberculosis* which make up about 10% of the genomic sequence, are unusual for their abundance and repetitive structure, suggesting *a priori* that they might play a role in the generation of antigenic variation [16]. These genes initially lacked homology to any known genes, but over time have been found in other *Mycobacterium* spp. and appear to be involved in virulence [42].

Another intriguing example of the detection of an unexpected (statistically unlikely) pattern is the discovery of sequences capable of acting as "molecular switches". Rapid, reversible genetic change (phenotypic variation or phase variation) is now known to be mediated by a number of different molecular mechanisms including slipped-strand mispairing, site-specific DNA rearrangement and DNA shuffling [30, 74]. These categories of mechanisms for genetic change represent a wide repertoire of potential mutations. Slipped-strand mispairing leads to small insertion–deletion events at short direct repeats, or microsatellites, while site-specific rearrangements are "triggered" by the presence of particular sequence motifs and DNA shuffling occurs when homologous section of DNA switch location or orientation within a genome. Such loci have been termed "contingency loci" [49] collectively. These "switches" are most often found in surface-exposed genes and genes that mediate key interactions with the host environment. They appear to have evolved both to provide genetic variation to evade host defenses and to facilitate rapid adaptation to different micro-niches within the host [4].

From a computational perspective, short direct repeats, or microsatellites, are the class of molecular switches that are most easily detected in new genome sequences. These loci, also known as simple sequence repeats (SSR), are composed of units of 1–6 bp. The high rate of slipped-strand mispairing at such loci results in the hypermutability of these sequences. Due to their ability to undergo rapid, reversible mutation they make highly efficient molecular switches [49] capable of altering the translation and transcription of genes, when found in coding regions or promoters, respectively. The discovery of a large number of such loci in the first sequenced bacterial genome, *H. influenzae* [33], has resulted in the search for such loci now being part of the routine analysis of novel pathogenic genomes. Chapter 8 gives a detailed account on methods for finding repeats in genome sequences.

The likelihood that novel SSRs are evolving in a nonrandom fashion can be evaluated by examining the total length (as longer loci are both more statistically improbable and more mutable [19]) as well as genetic context (i.e. the ability of an insertion or deletion event in an SSR to cause a frameshift within a known coding region or the location of an SSR in a promoter region) of each repeat. Candidate loci can be further evaluated through population-level surveys of variability in the "switch region" [46]. The best candidates can then be examined using empirical methods (such as knockout strains [33]) to explore their potential role in virulence and bacterial adaptation.

2.3 Comparative Genomic Methods of Detection

Good sampling of related genomes increases the power to detect differences and similarities between genomes. Given whole genomes, it is possible to search for genes shared between pathogens and nonpathogens, genes found only in pathogens, and genes that are never found in pathogens. This practice of “differential genome display” came into practice as soon as the complete genomes of pathogens and nonpathogens, i.e. *H. influenzae* and *E. coli*, became available [34]).

This approach works when pathogens share the genes responsible for a particular trait, but this is not always the case, as exemplified by the study of motility in Mycoplasmas. The Mycoplasmas are now one of the best phylogenetically represented groups of bacteria. They have minimal genome sizes, and yet display a variety of phenotypes, host preferences and degrees of pathogenicity. This group of bacteria is therefore ideal for comparative genomic studies in a phylogenetic framework. For example motility is a trait that has proven notoriously difficult to understand at the molecular level [35]. When a simple comparison between proteomes of motile and nonmotile Mycoplasmas failed to reveal candidate genes, the alternative was tested, i.e. that genes for motility were shared by motile and nonmotile species, but had been deactivated in nonmotile genomes [35]. Similarity searches revealed candidate genes that were better conserved among motile genomes than nonmotile genomes, suggesting relaxed selection acting on nonfunctional instances of these genes. Furthermore, the authors attempted to define a “required core set” of genes always found in full in motile genomes, but only ever found in part in nonmotile genomes. Both of these “filters” provided only circumstantial evidence of motility genes. Inspection of the 16S tree for the Mycoplasmas shows that motile and nonmotile species do not form two separate (monophyletic) evolutionary groups. The difficulties encountered in defining the genes responsible for motility may reflect the fact that motility has evolved independently more than once in Mycoplasmas [35]. A general overview of methods of comparative genomics is given in Chapter 37.

2.4 Taxonomically Restricted Genes (TRGs) and Orphans

TRGs are found in isolated lineages and it is therefore plausible that they are responsible for niche-specific traits, including those involved in virulence and pathogenicity. Among TRGs, “orphans” (predicted proteins that lack similarity to any known proteins) are of particular interest [78]. It has been claimed that the number of orphan genes discovered in complete genome sequences has been one of the biggest surprises of the genomic era so far [21]. Explanations for the large numbers of orphan genes in complete genomes include the suggestion that we have not sequenced enough genes from enough organisms to find other members of these gene families [73], that orphans may be noncoding sequences incorrectly annotated as predicted proteins [66] or that orphan genes belong to known gene families, but have not been identified as such owing to extensive sequence diversification [18]. An exciting frontier within the area of improved gene prediction is the use of transcriptomic and proteomic data to validate genomic annotations. For example, the first example of a "proteogenomic" approach to genome annotation found no evidence of expression of many smaller-than-average genes predicted using *in silico* methods alone, suggesting that many "orphans" do not encode real proteins [35]. Such empirical data is essential to correct errors of annotation and validate true orphans.

Despite the contribution of undersampling and error to the numbers of orphans found in public databases, the number of bacterial orphans appears to be increasing [78]. There is also mounting evidence that a proportion of orphans are real genes [18, 65]. For example, evidence of transcription was obtained in a study of 19 out of 25 proposed orphan genes in the *E. coli* K12 MG1655 [2]. An interesting possibility is that orphans are real genes captured by bacterial isolates through horizontal transfer from phage (and plasmids). This possibility is discussed further in Section 4.1 [18].

3 Putting Isolates of Infectious Bacteria into a Phylogenetic Framework

As described above, the use of phylogenetic and population genetic analysis can be used to address a wide range of questions concerning the biology and control of infectious bacteria (Table 1). Phylogenies can be used to "place" new isolates of pathogens, identified by their 16S sequences, onto the tree of life [60]. They can also be used to track the emergence of hypervirulent or antibiotic-resistant isolates [25], to explore the origins of pathogens and place dates on their emergence [24] or to test whether specific traits have evolved more than once [35]. Comprehensive phylogenies encompassing enough strains to adequately represent the genetic diversity of a named species can be used to guide a variety of subsequent studies. For example, to select a maximally divergent set of strains to test for evidence of recombination [26] or to examine the potential phylogenetic conservation of candidate epitopes for the development of vaccines [55, 79].

One of the most interesting conclusions about the nature of pathogens to be drawn from evolutionary studies is the fact that pathogenicity has evolved multiple times. Pathogens are often found next to nonpathogens on the bacterial tree and closely related isolates can differ greatly in their degree and nature of pathogenicity. For example the 16S sequence of *Yersinia pestis* and *Yersinia pseudotuberculosis* are identical [72], and the former appears to have evolved from the latter only 1500–20 000 years ago [1]. Despite their recent divergence, these species have very different potentials to cause disease. While *Y. pestis* is infamous for causing devastating disease (The

Plague), its close relative *Y. pseudotuberculosis* is rarely capable of causing disease or death.

It has become increasingly clear that different sections of bacterial genomes evolve at different rates. Studies must be designed in such a way that the rate of evolution of a chosen "marker" matches the question asked and provides the correct level of discrimination. For example, to resolve relationships between distantly related bacteria the 16S rDNA marker is preferable [53] because it is universally conserved. However, 16S sequences are often monomorphic among closely related strains and are therefore unsuitable for population-level studies. For isolates with complete genomes, it is now possible to construct trees based on complete proteome comparisons – an approach which removes the requirement for all isolated to share the same sequences. For example, trees can be built using the numbers of shared genes found in pairwise comparisons among proteomes [68] or on frequencies of amino acid motifs ("words") [58].

For examining the relationships between isolates within a species, more rapidly evolving loci are required. There are now a variety of methods available that assay genetic diversity capable of distinguishing closely related isolates. For example, ribotyping has been used to validate the utility of bacterial epitopes for use in vaccine development [8]. Ribotyping and similar methods that depend on the digestion of DNA by restriction enzymes are excellent for economically processing large numbers of strains from a single species, but have the drawback of generating "anonymous" banding patterns that lead to ambiguities in scoring the similarities and differences between strains. MLST, as mentioned above, is probably now the most widely applied method.

In some cases, though, MLST is unable to distinguish between isolates under examination, especially isolates from recently emerged pathogens. In the case of *Y. pestis*, for example, fragments of five housekeeping loci were examined and all were found to be monomorphic [1]. The three main biovars (strain types) of *Y. pestis* can be distinguished by assaying markers evolving more rapidly than housekeeping genes. These include restriction fragment length polymorphisms (RFLP) associated with the IS100 insertion element [1] and variable number tandem repeats (VNTRs) [57]. Such repeats are composed of direct repeats of 7–25 nucleotides which change in length with high frequency. The usefulness of these loci is increased by the typing of several loci simultaneously. This method was originally developed to distinguish strains of *Bacillus anthracis*, the most genetically monomorphic pathogen described to date, and is now referred to as multiple locus VNTR analysis (MLVA) [37]. MLVA can be used to rapidly type large numbers of isolates at low cost. More importantly, the resulting strain profiles can be compared between laboratories. Along with MLVA, SNPs and insertion/deletion polymorphisms have also been used to distinguish nearly identical strains [59].

Understanding the evolutionary relationships between strains has many important applications, but is challenging given that there is tremendous genetic variation among bacteria and bacterial species are notoriously ill-defined entities [41]. It is well-known that there can be significant differences both in the total quantity and composition of DNA between even very closely related isolates. The surprising degree of diversity, revealed through genome sequencing, even between different strains of the same bacterial species, implies that single genome sequences can no longer be viewed as defining the genetic repertoire of named taxa, but rather as a sample of the genes potentially available to members of a given (often ill-defined) population. For example, the analysis by Welch and coworkers [76] on three genome sequences of *E. coli* revealed that only about 40% of all identified open reading frames were common to all three strains. This finding has motivated the genome sequencing of multiple strains for a number of different named species, often in the hope that this will facilitate the discovery of the genetic basis of variable phenotypes of specific ecological or clinical relevance (such as heightened virulence or antibiotic resistance).

In order to examine the detailed dynamics of genome divergence at this fine-scale phylogenetic level, it is necessary to compare these genomes as thoroughly as possible within a phylogenetic or population biology framework. Now that we have the genomes of many pathogens, it is becoming easier to apply MLST approaches, but MLST studies can also be undertaken without the benefit of a genome (as was the case with the early MLST schemes) if enough suitable housekeeping genes, with uniform levels of variability, can be selected.

The case of *S. aureus*, a “species” for which seven complete genome sequences are now available, clearly demonstrates the importance of detailed information on evolutionary relationships when dealing with a set of isolates. MLST data has revealed *S. aureus* to be a highly clonal species, and the seven sequenced strains correspond to three pairs (each pair being very closely related and belonging to the same clone) and a single more diverged strain [32]. In the absence of the MLST data and without a complete appreciation of the genetic diversity in this species, it might be reasonable to suppose that the more diverged strain is atypical in some way. The ability to place the sequenced strains in the context of the whole population reveals that they are not representative of the full diversity within the population.

Nevertheless, the genome data for *S. aureus* enable comparisons both at a very fine level (within a given clone) and at a relatively more diverged level, encompassing the variation of the named species. Comparisons of gene content between pairs of strains belonging to the same clone reveals that a significant fraction (about 2%) of the genome is lost or gained extremely rapidly. This results in marked clinical differences between strains which are essentially identical in terms of nucleotide sequence divergence. However, it appears that the rate of change in gene content decreases in proportion to sequence divergence when comparisons are made between strains clearly separated on the phylogenetic tree. This example illustrates how comparisons between genomes belonging to the same species are most powerful when used in conjunction with MLST and/or microarray data that can provide the "big picture" for the structure and diversity of a given population [32].

4 Mixing of Genetic Material among Bacteria

Bacteria were once thought to be clonal organisms that evolved without recourse to recombination of their genetic material with that of other individuals. In the case of truly clonal organisms, the reconstruction of evolutionary relationships using any molecule will, in theory, reconstruct the same evolutionary tree. This will hold true for all loci at which sufficient "phylogenetically informative" polymorphisms are detectable and in the absence of both selection and convergent evolution. Recombination, or the exchange of DNA between independent lineages, can obscure the phylogenetic signal of individual lineages and lead to the reconstruction of trees which do not reflect the true relationships between isolates, but rather the individual evolutionary history of the recombinant molecules being scrutinized.

The last decade has revolutionized our view of the stability and clonality of bacteria through many lines of evidence and it is now clear that bacteria can acquire new pieces of DNA from a variety of sources. Until the advent of complete genome sequencing, the extent of horizontal transfer and the resulting "mosaic" nature of bacterial genomes was largely unappreciated. DNA acquired from the environment, or from infection with phage and plasmids, can result in significant changes in genetic repertoires in a way that can have a significant impact on a bacterium's success in a particular niche or even allow it to change its niche completely [51]. Such changes can result in the emergence of new pathogens or in heightened pathogenicity of existing pathogens. The single most important result of horizontal gene transfer is the global spread of antibiotic resistance.

Genes obtained through horizontal gene transfer can often be recognized by their atypical base composition, sporadic distribution within a given phylogeny or landmark features. For example, horizontally acquired sequences containing *H. influenzae* uptake signal sequences (USSs) have been found in *Neisseria meningitidis* [39]. Rates of horizontal transfer largely depend on niche (i.e. access to foreign DNA) and the presence of internal sequences that promote exchange. The small genomes of intracellular pathogens have few molecular mechanisms that promote exchange and contain little foreign DNA, while species with larger genomes might have gained up to 20% of their total gene content from external sources [51].

In addition to new genes or gene islands being acquired by horizontal gene transfer, thus launching the bacteria into a new, potentially pathogenic lifestyle, homologous recombination can also result in metabolic genes from any given strain being replaced by orthologs of a "donor" strain. MLST data, being based on ubiquitous "core" genes and often used on large isolate collections, are ideal for examining the extent to which this process occurs. By examining the sequence changes between very closely related isolates, it is possible to estimate the relative contributions of point mutations and homologous recombination [69]. This approach has been used to estimate that alleles change up to 10-fold more frequently by recombination than by point mutation in some species (e.g. *S. pneumoniae* and *N. meningitidis*) whereas in other species, such as *S. aureus*, point mutation appears to play a more major role than recombination [27, 28]. The reasons for these differences are unclear but are probably related to both the biology (e.g. efficiency of transformation) and ecology of the organism (e.g. the probability of diverse lineages meeting in the wild).

The extent of recombination can also be assayed from MLST data sets by comparing phylogenetic trees from a small representative sample of isolates. The logic behind this approach is that a history of frequent recombination will result in discordant trees when different gene loci are compared. A statistical approach based on the maximum likelihood method of phylogenetic reconstruction has been used to demonstrate that trees built from different gene loci for *S. pneumoniae* and *N. meningitidis* are typically no more similar to each other than they are to random trees [27, 28]. Thus, this approach is consistent with the method based on comparing very closely related isolates outlined above and indicates that recombination has been so frequent in this species that there is no consistent phylogenetic signal. It therefore makes little sense in such extreme cases to attempt to reconstruct the evolutionary relatedness between diverged strains, as these will be reticulate (network-like) rather than tree-like.

4.1 The Importance of Phage and Plasmids

The importance of plasmids and phage in bacterial evolution and adaptation has long been recognized, but it is continuing to come to the fore as evidence for the ubiquity of horizontal transfer accumulates. While phage are viruses and therefore pathogens (or "predators") of bacteria, it is more difficult to explain the persistence of plasmids [6] although it is clear that both provide important mechanisms for genetic exchange within bacteria. The gene pool potentially harbored within phage and plasmid populations is significant [44]. For example, the sequencing of 27 phage genomes from a single host, *S. aureus*, revealed a minimal overlap in gene content [40]. The discovery of such a high percentage of novel genes suggests further that phage store a vast reservoir of genetic diversity [40].

It is well recognized that prophages contribute significantly to observed inter-strain differences by donating, among other things, genes related to pathogenicity [9]. In a comparison of two *Salmonella typhi* strains, 113 predicted proteins were found to be unique to one strain. Seventy-six of these were prophage genes [20], illustrating the impact prophages have on the genetic individuality of bacterial strains. Similarly, in the species *Streptococcus pyogenes* all major sequence gaps found in the alignments of "M" serotypes are traceable to phage integration events [67]. Further, the prophages found in *S. pyogenes* encode proven or suspected virulence factors, and therefore the diversity observed between strains is likely to be of clinical relevance [5]. In a global study of 115 bacterial genomes, including pathogens and nonpathogens, 190 prophages were identified [10]. These prophages were predominantly found in pathogens. Several of them encoded disease-modifying factors found in species such as *Vibrio cholerae*, *E. coli* 0157 and *Corynebacterium diptheriae*. In some cases, including *S. aureus*, *S. pyogenes* and *Salmonella* spp., the genome contains multiple prophages, each of which contributes incrementally to virulence [10].

It has recently been suggested that phage contribute large numbers of orphan genes to bacteria [18]. It has been found that orphan genes have certain characteristics that differentiate them from other genes. Orphans are significantly shorter than native genes and are A + T-rich when contrasted with the rest of the genome [18]. These characteristics have led to the belief that phage may be responsible for the generation of bacterial orphans. Phage encode short A + T-rich genes [54]. The dinucleotide frequencies of *E. coli* orphans and of phage known to infect *E. coli* were found to be similarly biased in contrast with native genes [18]. In addition the genetic diversity of phage has been poorly sampled. Hence, if orphan genes did originate from phage it should not come as a great surprise that, as yet, we have not found homologs [18].

A classic example of the pathogen-defining potential of phage-encoded virulence factors is found in *V. cholerae*. The genome sequence of *V. cholerae* Tor N16961 revealed a single copy of the cholera toxin (CT) genes, *ctxAB*. These genes are localized within the integrated genome of CTX$^{\phi}$, a temperate filamentous phage [75]. The receptor for entry of CTX$^{\phi}$ into the cell is the toxin-coregulated pilus (TCP). The TCP also represents the critical intestinal colonization factor of *V. cholerae* [45]. The genes involved in TCP assembly are part of a pathogenicity island that also includes a helicase-related protein

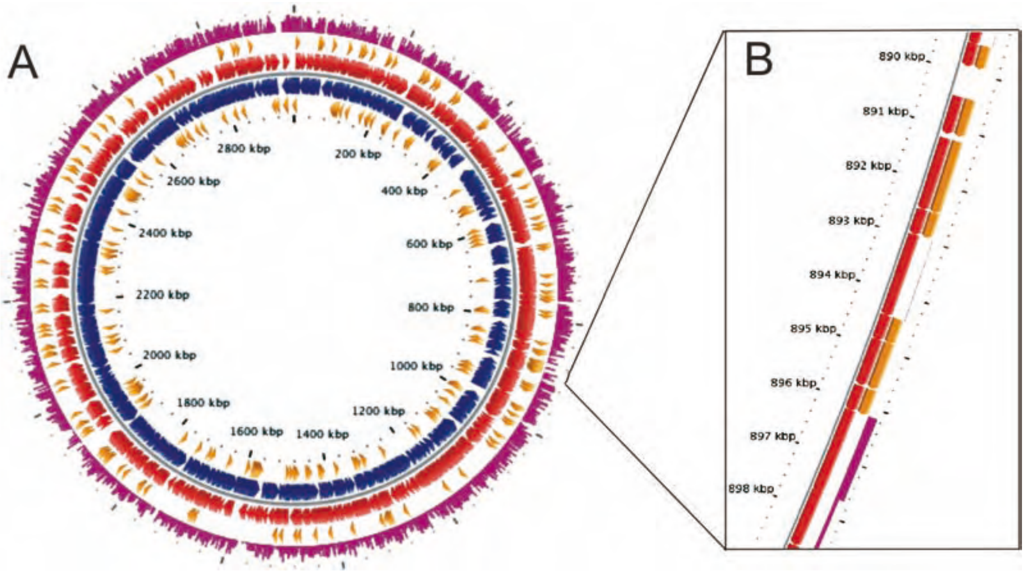


Figure 1 Circular representation of *V. cholerae* chromosome 1. This image was created using the program CGView [71] and data from the OrphanMine orphan gene database [78]. (A) Inner circle: predicted orphans on the negative strand. Second circle: predicted genes on the negative strand. Third circle: predicted genes on the positive strand. Fourth circle: predicted orphans on the positive strand. Fifth circle: representation of the number of bacterial species (from a total of 121 species) with hits to the associated gene. (B) The TCP Gene Cluster. Genes believed to be orphans prior to the sequencing of *V. fischeri* are shown in orange.

and a transcriptional activator that both share homology with bacteriophage proteins [31]. The TCP cluster of genes is shown in Figure 1. The majority of the genes located in the TCP cluster were long believed to be orphans until orthologs to many of these genes were found in the genome of the symbiotic bacterium of squid, *Vibrio fischeri* [63]. This surprising finding is made more intriguing by the suggestion that this region is native to *V. fischeri* but was acquired recently by *V. cholerae* from a low-GC genome such as *V. fischeri* [63].

This example illustrates how genes, annotated as orphans and derived from phage, can be responsible for important biological phenotypes that play a major role in the lifestyle of an organism. By sequencing more closely related genomes we will be able to "home" more of these taxonomically restricted put phenotypically relevant genes into gene families and in doing so, gain greater understanding of the evolution of these genomes. The discovery of genes associated with virulence in a commensal organism has more profound implications for the evolution of "virulence" factors and the role these genes play in nature. Although long-studied through the perspective of relevance to human health, it is possible that many of these genes carry out functions unrelated to pathogenicity in the environment. Perhaps this implies that virulence itself does not always evolve as a selected trait, but is an (unfortunate) byproduct which results from certain combinations of hitherto 'blameless' genes which occasionally arise in an ecological setting conducive to the onset of disease.

5 Coevolution of Infectious Bacteria with Their Hosts

The coevolution of infectious bacteria with their hosts is just beginning to inform medical science and yet is essential to understand when attempting to interpret levels of variation found between isolates [80]. For example, it is crucial to understand pathogen ecology and distribution with respect to its host(s) since shifts in either can lead to a jump between hosts, a phenomenon increasingly being seen in the case of emerging diseases. Here, we discuss two aspects of the coevolution of pathogens with their hosts that have a significant impact on the ability of pathogens to cause disease and which have been revolutionized through the sequencing of genomes. These are the coevolution (reduction) of metabolic capacity in pathogens associated with their hosts over long evolutionary time periods and the evolution of mechanisms for rapidly generating genetic variation that arise as a direct result of having to survive within the ever changing and hostile environment of a host.

5.1 Reconstructing Metabolic Pathways

As described above, while many bacterial strains become pathogens, or gain heightened pathogenicity, by acquiring new genes, the long-term evolution of many pathogens involves a loss of genes. The size of a bacterial genome can range from less than 0.5 to almost 10 million base pairs and this variation depends on the number of genes required to live in a specific niche. Larger genomes have more complex metabolic capacities as well as a high degree of redundancy in their pathways. Small genomes reflect a degree of specialization, with the smallest genomes belonging to obligate intracellular parasites.

The metabolic potential of any pathogen is largely a product of the amount of time it has coevolved with a particular host(s). The most interesting insights into this phenomenon are gained from the examination of the metabolic pathways lost in genomes undergoing reductive evolution (reduction of genome size). As pathogens become more dependent on their hosts over long evolutionary time periods, they lose the need to maintain a complete set of metabolic capabilities. If a given pathway is lost, it is likely that the corresponding metabolite is being acquired directly from the host.

Snapshots of this process in different species have been captured by the sequencing of genomes. We now have the complete genomes of several species that have experienced different degrees of reductive evolution. These include the insect symbionts Buchnera and the intracellular pathogens like Mycoplasmas, which are of special interest because they represent "minimal" genomes. The larger genomes of species of Rickettsia and Mycobacteria are still undergoing reductive evolution, as is evidenced by the relative sizes of the repertoire of pseudogenes (nonfunctional, or "dead" genes) that are currently present in these genomes. It is clear that the route to a reduced genome can vary significantly among different organisms and very different gene complements can be successfully adopted.

Detailed metabolic knowledge of an organism can have important practical applications, for example in the study of fastidious and unculturable pathogens in the laboratory. Computer modeling of metabolic networks in the *Tropheryma whipplei* genome led to the design of a medium that allowed *T. whipplei* strains, a species previously only culturable in the presence of human fibroblast cells, to grow in cell-free culture [61]. The success of this approach has led the authors to develop the MetaGrowth Knowledgebase, an online source of empirical and *in silico* evidence for the culture of species, which will hopefully lead to better culture conditions for species including *Coxiella burnetii*, *Mycobacterium leprae*, *Rickettsia prowazekii*, *Rickettsia conorii* and *Treponema pallidum* [52]. The biochemical network view of host–pathogen interaction is discussed in Chapter 23.

5.2 The Genetic Arms Race between Pathogen and Host

The host has a variety of mechanisms for defending itself against pathogens and as a result host–pathogen relationships are associated with some of the highest known selective pressures found in nature. The analysis of genomic sequences has provided new insights into the mechanisms by which bacteria invade and survive in host environments, and these often involve the rapid generation of genetic variability. In addition, genomic sequencing is providing detailed evidence to support one of the most fascinating observations to come from the study of bacterial mechanisms of generating genetic diversity, namely that bacteria (and other organisms) have evolved "the ability to evolve" [11].

Random genetic mutation is the "engine" of evolution producing variation that can be the source of new adaptive genotypes. Bacteria have two strategies for elevating the number of beneficial mutations generated within a given period of time. The first is to raise global mutation rates and the second is to evolve regions of localized hypermutation [48]. "Mutators" usually arise through a genetic mutation in a gene related to DNA replication or repair.

Damage to such genes can increase the number of errors made during chromosomal replication or decrease the rate at which errors are repaired leading to the fixation of higher than average numbers of mutations. “Mutators” have been associated with the emergence of hypervirulence and antibiotic resistance [14].

Raising global mutation rates above background rates increases the probability of a beneficial mutation in the short term (because there are overall more mutations occurring), but will have a damaging effect on an organism’s fitness over the longer term as the vast majority of mutations are deleterious [47]. To avoid this problem, many pathogenic bacteria have evolved a different strategy of increasing rates of evolution. This second strategy involves the ability to generate large numbers of mutations at specific locations within their genomes.

Phase variation, or phenotypic switching, is a widespread adaptive strategy among pathogens [30]. It is well known that DNA sequences of different compositions can have varying mutation rates. Bacteria have capitalized on this phenomenon to evolve a variety of “molecular switches” that generate rapid and reversible change. For example, as described above, candidate SSR contingency loci are straightforward to extract from DNA and with the sequencing of a variety of pathogen genomes we are beginning to appreciate the extent to which these phase-variable loci are distributed among different taxa. It is also becoming clear that contingency loci have evolved many times during the course of bacterial evolution (Figure 2). Of the first 220 sequenced isolates of bacteria, to the best of our knowledge, at least 30 isolates are reported to possess putative or known SSR contingency loci. These isolates belong to 30 species of 18 genera found in five divisions of bacteria.

6 Conclusions

The importance of an evolutionary perspective in the study of infectious bacteria is coming into the fore in the era of genomics. The analysis of genome sequences is elevating our general appreciation of the tremendous amount of genetic diversity harbored in the bacterial gene pool and is elucidating phenomena of specific importance to clinicians. These include the rise in antibiotic resistance and emerging diseases. The tremendous potential of bacteria to undergo adaptive evolutionary change can confound the development and application of widely efficacious therapies and make pathogens formidable foes of humans which can have devastating consequences for human health. Collaborations between evolutionists, bioinformaticians, clinicians and those generating complete genome sequences should be fostered as this approach will grow in importance as we continue to sequence more bacterial genomes

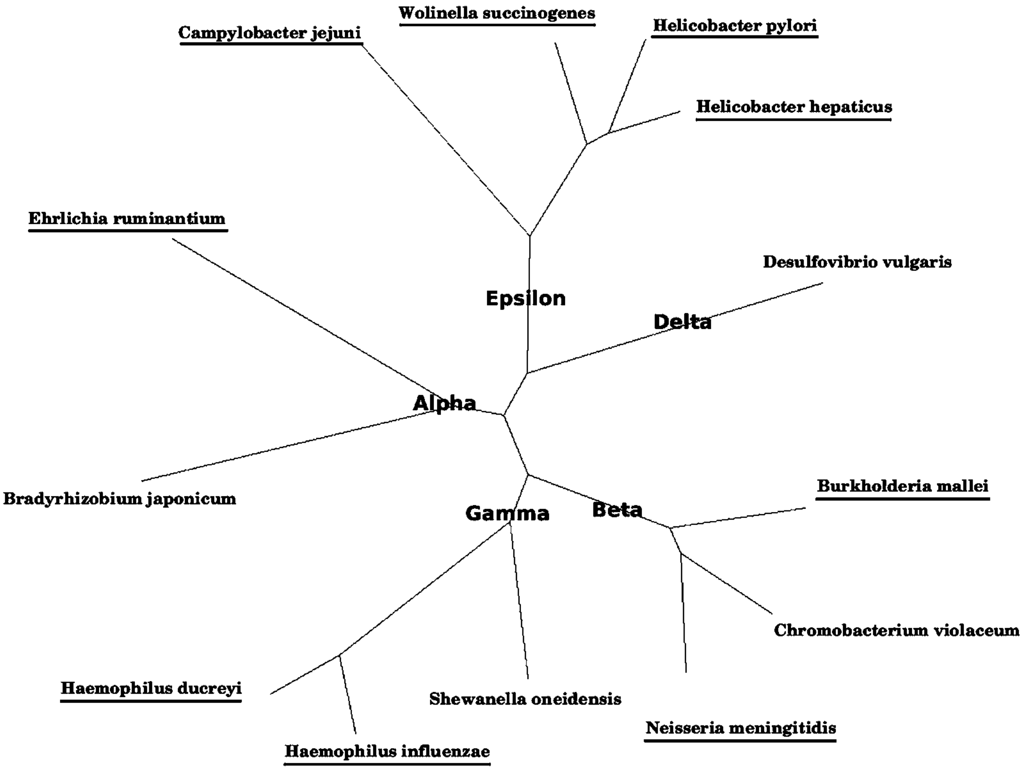


Figure 2 A phylogenetic tree of selected species of proteobacteria demonstrates the independent evolution of phase-variation among commensals and pathogens. Characterized and putative SSR contingency loci have been found in the underlined species. These species are interspersed among free-living environmental species (nonunderlined species). This unrooted neighbor-joining tree was constructed using the Jukes–Cantor method from pre-aligned 16S ribosomal sequences downloaded from the RDP [15].

and generate phylogenetic data sets for large sets of isolates. The study of pathogens using molecular evolutionary theory and bioinformatics is fuelling the emergence of the multidisciplinary field of evolutionary pathogenomics.

The wealth of genomic and phylogenetic data presents vast opportunities for the study of infectious bacteria. Along with these opportunities come challenges, including the need for better methods, databases and tools for mining data from collections of genomes in a phylogenetic context [29]. There is also a pressing need for additional information on the specific phenotypes associated with clinical isolates, improved sampling of natural isolates, and targeted sequencing of nonpathogens. Understanding the extent, fluidity and significance of the bacterial gene pool is one of the grand challenges for researchers working in this field. The most powerful data sets will be those with a phylogenetic tree describing the entire genetic diversity of a species (i.e. thousands of isolates), 10–30 complete genomes to allow detailed genomic analyses coupled with microarray-based typing studies of hundreds of strains. Such data sets are for the future, but at the present rate of advance are unlikely to be too far over the horizon.

References

1 ACHTMAN, M., K. ZURTH, G. MORELLI, G. TORREA, A. GUIYOULE AND E. CARNIEL. 1999. *Yersinia pestis*, the cause of plague, is a recently emerged clone of *Yersinia pseudotuberculosis*. Proc. Natl Acad. Sci. USA **96**: 14043–8.

2 ALIMI, J. P., O. POIROT, F. LOPEZ AND J. M. CLAVERIE. 2000. Reverse transcriptase-polymerase chain reaction validation of 25 "orphan" genes from *Escherichia coli* K-12 MG1655. Genome Res. **10**: 959–66.

3 ALTSCHUL, S. F., W. GISH, W. MILLER, E. W. MYERS AND D. J. LIPMAN. 1990. Basic local alignment search tool. J. Mol. Biol. **215**: 403–10.

4 BAYLISS, C. D., D. FIELD AND E. R. MOXON. 2001. The simple sequence contingency loci of *Haemophilus influenzae* and *Neisseria meningitidis*. J. Clin. Invest. **107**: 657–62.

5 BERES, S. B., G. L. SYLVA, K. D. BARBIAN, et al. 2002. Genome sequence of a serotype M3 strain of group A Streptococcus: phage-encoded toxins, the high-virulence phenotype, and clone emergence. Proc. Natl Acad. Sci. USA **99**: 10078–83.

6 BERGSTROM, C. T., M. LIPSITCH AND B. R. LEVIN. 2000. Natural selection, infectious transfer and the existence conditions for bacterial plasmids. Genetics **155**: 1505–19.

7 BERNAL, A., U. EAR AND N. KYRPIDES. 2001. Genomes OnLine Database (GOLD): a monitor of genome projects world-wide. Nucleic Acids Res. **29**: 126–7.

8 BOLDUC, G. R., V. BOUCHET, R. Z. JIANG, J. GEISSELSODER, Q. C. TRUONG-BOLDUC, P. A. RICE, S. I. PELTON AND R. GOLDSTEIN. 2000. Variability of outer membrane protein P1 and its evaluation as a vaccine candidate against experimental otitis media due to nontypeable *Haemophilus influenzae*: an unambiguous, multifaceted approach. Infect. Immun. **68**: 4505–17.

9 BRUSSOW, H., C. CANCHAYA AND W. D. HARDT. 2004. Phages and the evolution of bacterial pathogens: from genomic rearrangements to lysogenic conversion. Microbiol. Mol. Biol. Rev. **68**: 560–602.

10 CANCHAYA, C., G. FOURNOUS AND H. BRUSSOW. 2004. The impact of prophages on bacterial chromosomes. Mol. Microbiol. **53**: 9–18.

11 CAPORALE, L. H. 2003. Natural selection and the emergence of a mutation phenotype: an update of the evolutionary synthesis considering mechanisms that affect genome variation. Annu. Rev. Microbiol. **57**: 467–85.

12 CHAN, M. S., M. C. MAIDEN AND B. G. SPRATT. 2001. Database-driven multi locus sequence typing (MLST) of bacterial pathogens. Bioinformatics **17**: 1077–83.

13 CHEN, L., J. YANG, J. YU, Z. YAO, L. SUN, Y. SHEN AND Q. JIN. 2005. VFDB: a reference database for bacterial virulence factors. Nucleic Acids Res. **33**: D325–8.

14 CHOPRA, I., A. J. O'NEILL AND K. MILLER. 2003. The role of mutators in the emergence of antibiotic-resistant bacteria. Drug Resist. Updat. **6**: 137–45.

15 COLE, J. R., B. CHAI, R. J. FARRIS, Q. WANG, S. A. KULAM, D. M. MCGARRELL, G. M. GARRITY AND J. M. TIEDJE. 2005. The Ribosomal Database Project (RDP-II): sequences and tools for high-throughput rRNA analysis. Nucleic Acids Res. **33**: D294–6.

16 COLE, S. T., R. BROSCH, J. PARKHILL, et al. 1998. Deciphering the biology of *Mycobacterium tuberculosis* from the

complete genome sequence. Nature **393**: 537–44.

17 COOPER, J. E. AND E. J. FEIL. 2004. Multilocus sequence typing – what is resolved? Trends Microbiol. **12**: 373–7.

18 DAUBIN, V. AND H. OCHMAN. 2004. Bacterial genomes as new gene homes: The genealogy of ORFans in *E. coli*. Genome Res. **14**: 1036–42.

19 DE BOLLE, X., C. D. BAYLISS, D. FIELD, T. VAN DE VEN, N. J. SAUNDERS, D. W. HOOD AND E. R. MOXON. 2000. The length of a tetranucleotide repeat tract in *Haemophilus influenzae* determines the phase variation rate of a gene with homology to type III DNA methyltransferases. Mol. Microbiol. **35**: 211–22.

20 DENG, W., V. BURLAND, G. PLUNKETT, 3RD, et al. 2002. Genome sequence of Yersinia pestis KIM. J. Bacteriol. **184**: 4601–11.

21 DOOLITTLE, R. F. 2002. Biodiversity: microbial genomes multiply. Nature **416**: 697–700.

22 EISEN, J. A. AND C. M. FRASER. 2003. Phylogenomics: intersection of evolution and genomics. Science **300**: 1706–7.

23 EPPINGER, M., C. BAAR, G. RADDATZ, D. H. HUSON AND S. C. SCHUSTER. 2004. Comparative analysis of four Campylobacterales. Nat. Rev. Microbiol. **2**: 872–85.

24 FALUSH, D., C. KRAFT, N. S. TAYLOR, P. CORREA, J. G. FOX, M. ACHTMAN AND S. SUERBAUM. 2001. Recombination and mutation during long-term gastric colonization by *Helicobacter pylori*: estimates of clock rates, recombination size, and minimal age. Proc. Natl Acad. Sci. USA **98**: 15056–61.

25 FEIL, E. J. AND M. C. ENRIGHT. 2004. Analyses of clonality and the evolution of bacterial pathogens. Curr. Opin. Microbiol. **7**: 308–13.

26 FEIL, E. J., E. C. HOLMES, D. E. BESSEN, et al. 2001. Recombination within natural populations of pathogenic bacteria: short-term empirical estimates and long-term phylogenetic consequences. Proc. Natl Acad. Sci. USA **98**: 182–7.

27 FEIL, E. J., M. C. MAIDEN, M. ACHTMAN AND B. G. SPRATT. 1999. The relative contributions of recombination and mutation to the divergence of clones of *Neisseria meningitidis*. Mol. Biol. Evol **16**: 1496–502.

28 FEIL, E. J., J. M. SMITH, M. C. ENRIGHT AND B. G. SPRATT. 2000. Estimating recombinational parameters in *Streptococcus pneumoniae* from multilocus sequence typing data. Genetics **154**: 1439–50.

29 FIELD, D., E. FEIL AND G. WILSON. 2005. Databases and software for the comparisons of prokaryotic genomes. Microbiology **151**: 2125–32.

30 HALLET, B. 2001. Playing Dr Jekyll and Mr Hyde: combined mechanisms of phase variation in bacteria. Curr. Opin. Microbiol. **4**: 570–81.

31 HEIDELBERG, J. F., J. A. EISEN, W. C. NELSON, et al. 2000. DNA sequence of both chromosomes of the cholera pathogen *Vibrio cholerae*. Nature **406**: 477–83.

32 HOLDEN, M. T., E. J. FEIL, J. A. LINDSAY, et al. 2004. Complete genomes of two clinical *Staphylococcus aureus* strains: evidence for the rapid evolution of virulence and drug resistance. Proc. Natl Acad. Sci. USA **101**: 9786–91.

33 HOOD, D. W., M. E. DEADMAN, M. P. JENNINGS, M. BISERCIC, R. D. FLEISCHMANN, J. C. VENTER AND E. R. MOXON. 1996. DNA repeats identify novel virulence genes in *Haemophilus influenzae*. Proc. Natl Acad. Sci. USA **93**: 11121–5.

34 HUYNEN, M. A., Y. DIAZ-LAZCOZ AND P. BORK. 1997. Differential genome display. Trends Genet. **13**: 389–90.

35 JAFFE, J. D., N. STANGE-THOMANN, C. SMITH, et al. 2004. The complete genome and proteome of *Mycoplasma mobile*. Genome Res. **14**: 1447–61.

36 JOLLEY, K. A., E. J. FEIL, M. S. CHAN AND M. C. MAIDEN. 2001. Sequence type analysis and recombinational tests (START). Bioinformatics **17**: 1230–1.

37 KEIM, P., L. B. PRICE, A. M. KLEVYTSKA, K. L. SMITH, J. M. SCHUPP, R. OKINAKA,

P. J. Jackson and M. E. Hugh-Jones. 2000. Multiple-locus variable-number tandem repeat analysis reveals genetic relationships within *Bacillus anthracis*. J. Bacteriol. **182**: 2928–36.

38 Kimura, M. 1968. *The Neutral Theory of Molecular Evolution*. Cambridge University Press, Cambridge.

39 Kroll, J. S., K. E. Wilks, J. L. Farrant and P. R. Langford. 1998. Natural genetic exchange between *Haemophilus* and *Neisseria*: intergeneric transfer of chromosomal genes between major human pathogens. Proc. Natl Acad. Sci. USA **95**: 12381–5.

40 Kwan, T., J. Liu, M. Dubow, P. Gros and J. Pelletier. 2005. The complete genomes and proteomes of 27 *Staphylococcus aureus* bacteriophages. Proc. Natl Acad. Sci. USA **102**: 5174–9.

41 Lan, R. and P. R. Reeves. 2000. Intraspecies variation in bacterial genomes: the need for a species genome concept. Trends Microbiol. **8**: 396–401.

42 Li, Y., E. Miltner, M. Wu, M. Petrofsky and L. E. Bermudez. 2005. A *Mycobacterium avium* PPE gene is associated with the ability of the bacterium to grow in macrophages and virulence in mice. Cell Microbiol. **7**: 539–48.

43 Maiden, M. C., J. A. Bygraves, E. Feil, et al. 1998. Multilocus sequence typing: a portable approach to the identification of clones within populations of pathogenic microorganisms. Proc. Natl Acad. Sci. USA **95**: 3140–5.

44 Mann, N. H. 2005. The third age of phage. PLoS Biol. **3**: e182.

45 Manning, P. A. 1997. The tcp gene cluster of *Vibrio cholerae*. Gene **192**: 63–70.

46 Martin, P., T. van de Ven, N. Mouchel, A. C. Jeffries, D. W. Hood and E. R. Moxon. 2003. Experimentally revised repertoire of putative contingency loci in *Neisseria meningitidis* strain MC58: evidence for a novel mechanism of phase variation. Mol. Microbiol. **50**: 245–57.

47 Matic, I., F. Taddei and M. Radman. 2004. Survival versus maintenance of genetic stability: a conflict of priorities during stress. Res. Microbiol. **155**: 337–41.

48 Metzgar, D. and C. Wills. 2000. Evolutionary changes in mutation rates and spectra and their influence on the adaptation of pathogens. Microbes Infect. **2**: 1513–22.

49 Moxon, E. R., P. B. Rainey, M. A. Nowak and R. E. Lenski. 1994. Adaptive evolution of highly mutable loci in pathogenic bacteria. Curr. Biol. **4**: 24–33.

50 Nielsen, R. and M. J. Hubisz. 2005. Evolutionary genomics: detecting selection needs comparative data. Nature **433**: E6; discussion E7–8.

51 Ochman, H., J. G. Lawrence and E. A. Groisman. 2000. Lateral gene transfer and the nature of bacterial innovation. Nature **405**: 299–304.

52 Ogata, H. and J. M. Claverie. 2005. Metagrowth: a new resource for the building of metabolic hypotheses in microbiology. Nucleic Acids Res. **33**: D321–4.

53 Olsen, G. J. and C. R. Woese. 1993. Ribosomal RNA: a key to phylogeny. FASEB J. **7**: 113–23.

54 Pedulla, M. L., M. E. Ford, J. M. Houtz, et al. 2003. Origins of highly mosaic mycobacteriophage genomes. Cell **113**: 171–82.

55 Pizza, M., V. Scarlato, V. Masignani, et al. 2000. Identification of vaccine candidates against serogroup B meningococcus by whole-genome sequencing. Science **287**: 1816–20.

56 Plotkin, J. B., J. Dushoff and H. B. Fraser. 2004. Detecting selection using a single genome sequence of *M. tuberculosis* and *P. falciparum*. Nature **428**: 942–5.

57 Pourcel, C., F. Andre-Mazeaud, H. Neubauer, F. Ramisse and G. Vergnaud. 2004. Tandem repeats analysis for the high resolution phylogenetic analysis of *Yersinia pestis*. BMC Microbiol. **4**: 22.

58 Qi, J., H. Luo and B. Hao. 2004. CVTree: a phylogenetic tree reconstruction tool based on whole genomes. Nucleic Acids Res. **32**: W45–7.

59 READ, T. D., S. L. SALZBERG, M. POP, et al. 2002. Comparative genome sequencing for discovery of novel polymorphisms in *Bacillus anthracis*. Science **296**: 2028–33.

60 RELMAN, D. A., T. M. SCHMIDT, R. P. MACDERMOTT AND S. FALKOW. 1992. Identification of the uncultured bacillus of Whipple's disease. N. Engl. J. Med **327**: 293–301.

61 RENESTO, P., N. CRAPOULET, H. OGATA, B. LA SCOLA, G. VESTRIS, J. M. CLAVERIE AND D. RAOULT. 2003. Genome-based design of a cell-free culture medium for *Tropheryma whipplei*. Lancet **362**: 447–9.

62 ROCHA, E. P. C., J. MAYNARD SMITH, L. D. HURST, M. T. G. HOLDEN, J. E. COOPER, N. H. SMITH AND E. J. FEIL. 2005. Comparisons of dN/dS are time-dependent for closely related bacterial genomes. J. Theor. Biol. **239**:226–35.

63 RUBY, E. G., M. URBANOWSKI, J. CAMPBELL, et al. 2005. Complete genome sequence of *Vibrio fischeri*: a symbiotic bacterium with pathogenic congeners. Proc. Natl Acad. Sci. USA **102**: 3004–9.

64 SCHMIDT, H. AND M. HENSEL. 2004. Pathogenicity islands in bacterial pathogenesis. Clin. Microbiol. Rev. **17**: 14–56.

65 SHMUELY, H., E. DINITZ, I. DAHAN, J. EICHLER, D. FISCHER AND B. SHAANAN. 2004. Poorly conserved ORFs in the genome of the archaea *Halobacterium* sp NRC-1 correspond to expressed proteins. Bioinformatics **20**: 1248–53.

66 SKOVGAARD, M., L. J. JENSEN, S. BRUNAK, D. USSERY AND A. KROGH. 2001. On the total number of genes and their length distribution in complete microbial genomes. Trends Genet. **17**: 425–8.

67 SMOOT, J. C., K. D. BARBIAN, J. J. VAN GOMPEL, et al. 2002. Genome sequence and comparative microarray analysis of serotype M18 group A Streptococcus strains associated with acute rheumatic fever outbreaks. Proc. Natl Acad. Sci. USA **99**: 4668–73.

68 SNEL, B., P. BORK AND M. A. HUYNEN. 1999. Genome phylogeny based on gene content. Nat. Genet. **21**: 108–10.

69 SPRATT, B. G., W. P. HANAGE AND E. J. FEIL. 2001. The relative contributions of recombination and point mutation to the diversification of bacterial clones. Curr. Opin. Microbiol. **4**: 602–6.

70 SPRATT, B. G., W. P. HANAGE, B. LI, D. M. AANENSEN AND E. J. FEIL. 2004. Displaying the relatedness among isolates of bacterial species – the eBURST approach. FEMS Microbiol. Lett. **241**: 129–34.

71 STOTHARD, P. AND D. S. WISHART. 2005. Circular genome visualization and exploration using CGView. Bioinformatics **21**: 537–9.

72 TREBESIUS, K., D. HARMSEN, A. RAKIN, J. SCHMELZ AND J. HEESEMANN. 1998. Development of rRNA-targeted PCR and *in situ* hybridization with fluorescently labelled oligonucleotides for detection of *Yersinia* species. J. Clin. Microbiol. **36**: 2557–64.

73 UNGER, R., S. ULIEL AND S. HAVLIN. 2003. Scaling law in sizes of protein sequence families: from super-families to orphan genes. Proteins **51**: 569–76.

74 VAN DER WOUDE, M. W. AND A. J. BAUMLER. 2004. Phase and antigenic variation in bacteria. Clin. Microbiol. Rev. **17**: 581–611.

75 WALDOR, M. K. AND J. J. MEKALANOS. 1996. Lysogenic conversion by a filamentous phage encoding cholera toxin. Science **272**: 1910–4.

76 WELCH, R. A., V. BURLAND, G. PLUNKETT, 3RD, et al. 2002. Extensive mosaic structure revealed by the complete genome sequence of uropathogenic *Escherichia coli*. Proc. Natl Acad. Sci. USA **99**: 17020–4.

77 WHEELER, D. L., T. BARRETT, D. A. BENSON, et al. 2005. Database resources of the National Center for Biotechnology Information. Nucleic Acids Res. **33**: D39–45.

78 WILSON, G. A., N. BERTRAND, Y. PATEL, J. B. HUGHES, E. J. FEIL AND F. D. 2005. Orphans as taxonomically restricted and ecologically important genes. Microbiology **151**: 2499–501.

79 WIZEMANN, T. M., J. H. HEINRICHS, J. E. ADAMOU, et al. 2001. Use of a whole genome approach to identify vaccine molecules affording protection against *Streptococcus pneumoniae* infection. Infect. Immun. **69**: 1593–8.

80 Woolhouse, M. E., J. P. Webster, E. Domingo, B. Charlesworth and B. R. Levin. 2002. Biological and biomedical implications of the co-evolution of pathogens and their hosts. Nat. Genet. **32**: 569–77.

81 Zwick, M. E., F. McAfee, D. J. Cutler, T. D. Read, J. Ravel, G. R. Bowman, D. R. Galloway and A. Mateczun. 2005. Microarray-based resequencing of multiple *Bacillus anthracis* isolates. Genome Biol. **6**: R10.

Part 10
Basic Bioinformatics Technologies

42
Integrating Biological Databases

Zoé Lacroix, Bertram Ludäscher and Robert Stevens

1 Biological Resources

Biological resources useful to the scientists include a variety of data sources and applications. These resources are made publicly available to the researcher on the web, or are privately accessed through subscriptions or within the scope of collaborations. The number of these resources is overwhelming and increases significantly each year. In 2005, the Molecular Biology Database Collection listed 719 public data sources [33] – a 31% increase since the previous year and 351% since the first compiled list in 1999 [15]. The thousands of data sources providing biological information overlap: multiple data sources may provide information about a scientific object, e.g. a nucleotide sequence, and thus share entries. When these data sources provide information about the same scientific objects, e.g. a gene, their data differ dramatically with respect to (i) organization ("How is the scientific object represented?"), (ii) format [ASCII, eXtensible Markup Language (XML), etc.], (iii) number of entries (the instances of the scientific object entered in the data source), (iv) characterization ("What information related to the scientific object is provided?") and (v) data quality ("What level of curation is achieved?"). As scientific objects are related by various scientifically meaningful relationships, e.g. a gene *codes for* a protein, the biological data sources providing information about those objects are often intertwined with links that capture those relationships, expressing them through hyperlinks, indices or generated by applications.

Scientific reasoning in bioinformatics relies on complex scientific protocols partially or completely involving digital data sets retrieved from public or

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

private data sources. The digital parts of scientific protocols include data collection from various sources, data analysis – using a wide range of applications – and data transformation, including traditional database operations such as selections, projections, and joins. The execution – automated or not – of a single protocol may thus involve multiple resources including data sources and applications [77]. Integrating these resources and the data flows resulting from the execution of scientific protocols is critical to scientific discovery. Other areas in which scientific data integration is the core technology include the design of scientific knowledge repositories and the representation of systems biology and scientific pathways. Systems biology relates to different types of biological information, such as DNA, RNA, protein, protein interactions, cells, tissues, etc., together with their individual elements, e.g. specific genes or proteins, and the relationships of these with respect to one another and the elements of other types. The aim of systems biology is to integrate all of this information to obtain a view or model of the system as a whole. Similarly, molecular interaction networks, including metabolic pathways, regulatory pathways and molecular complexes, aim at modeling the complex interactions between various scientific objects and the systems that support them aim at providing access to information relevant to these entities and their relationships.

The problems pertaining to scientific data integration are inherently complex and challenging. Scientific data integration relies on the integration of various systems often developed on different platforms, with different operating systems. These primary differences already raise various operability problems of communications between the integrated resources and the integrating system. Then, scientific integration requires the integration of the data themselves – a task that requires the integrating system to access and "understand" the data. Accessing and understanding the data requires from the integration system the ability to express (retrieval or database) queries to the data source so that the integrated resource provides the expected data and to map the data to the integrated data representation. In addition, scientific integration requires from the integration system to provide all analysis, transformation and, sometimes, visualization tools that are available to the scientist. In fact, to the scientist, data access (via a database or a retrieval query) and data analysis (via a tool) often are alike. Applications and data both play primary roles in the scientific data integration scenario. Data without suitable technology to access and analyze them are as meaningless as the technology without the data. For these reasons, a data integration approach should allow flexibility of use of the integrated data with available scientific tools.

Many approaches have been developed in the past to address these problems. Although they all offer benefits to the life scientist, they fail to provide a complete solution to the scientific integration problem. The main reason may reside in the legacy with which they have to deal. There are several communities that have developed integration technology, although not always primarily motivated by scientific integration challenges: (i) the database community which develops database management systems and focuses on the design of data definition languages to represent the data and query languages to access and transform the data, (ii) the computing community which develops approaches to facilitate interoperability of applications with distributed computing architectures, Common Object Request Broker Architecture (CORBA), the Grid, peer-to-peer (P2P) and web services, (iii) the knowledge community which develops intelligent-based approaches where integration is driven by a meaning expressed with Semantic Web or ontologies and achieved by intelligent agents, and (iv) the process community which aims at modeling business processes with workflows. As biological data integration involves a combination of data, computing, process and knowledge, we believe that a solution suitable to scientific discovery should benefit from all approaches. For these reasons, we present the main contributions from those domains that may shape in some extent the next generations of biological integration systems.

The chapter is organized as follows. Various data models are presented in Section 2. Section 3 successively addresses the problems of the mapping of the conceptual model to underlying models and introduces three traditional approaches to data integration: data warehouse, linked-based federation and mediation. The problem of application integration is addressed in Section 4. Section 5 is devoted to semantic issues related to data integration. In Section 6, we discuss scientific workflows, which aim at combining data access, integration, analysis and visualization into an integrated whole.

2 Data Modeling

An issue critical to data integration is related to data models. As integrated data are typically extracted from multiple resources, each having its own data organization, data model and data structure, reconciling these different modes of data modeling into a integrated model requires adequate understanding of these three layers of data modeling. Although the terminology to distinguish those three layers may differ in the literature, to clarify the discussion, we define them as follows.

- *Data organization* (or conceptual model) pertains to the way data is understood by the data providers. For example, GenBank is sequence-centric (all information the data source provides is organized with respect to the scientific object sequence).

- *Data model* is one among various data management definition languages offered (mostly) by the database community. They include relational model, nested-relational model, object-oriented model, XML, etc. Each database management system provides such a data model that captures the internal representation of the data in the system.
- *Data structure* (or schema) is the chosen structure with respect to the organization and model. Once a data model is selected, it offers multiple ways to represent internally the data. The data structure in the relational model is defined with respect to the tables (relations) and columns (attributes) of the relational schema. The data structure of the XML data model is characterized by a document type definition (DTD) or a XML schema.

The selection of each of these layers of data representation may affect significantly all aspects of data management including: data storage (ease of entering the data and efficiency of storage), data access (expressive power of query language), maintenance, etc.

2.1 Conceptual Model

The data organization may be represented with models designed not to manage the data, but to organize the data and represent their meaning. Such models include the Entity–Relationship (ER) model and the Unified Modeling Language (UML), respectively developed by the database community and by the programming languages community. These models provide abstract data representations that are not used to actually format the data (entities). Conceptual models represent classes of scientific objects such as *gene* or *sequence*, and their relationships and processes, but they do not represent the instances of these scientific objects. Instances will be represented with respect to the data structure, the third layer of data modeling.

2.1.1 **ER**

Schema diagrams were formalized in the 1960s by Charles Bachman who used rectangles to denote record types and directed edges (arrows) from one record type to another to denote relationships between instances of the two types [2, 3]. Peter Chen later used a similar formalism called ER where rectangles formalize entity types (or entities) and edges labeled with a diamond formalize relationship types (or relationships) between entities [21]. Although relationships in the ER are not directed, their connectivity – the constraint on the number of instances of entities they may connect – is specified with a number, e.g. 1, or a letter, e.g. M, that, respectively, expresses that a single instance or multiple instances of the entity may be linked with the relationship. Both entities and relationships may have attributes to describe their characteristics.

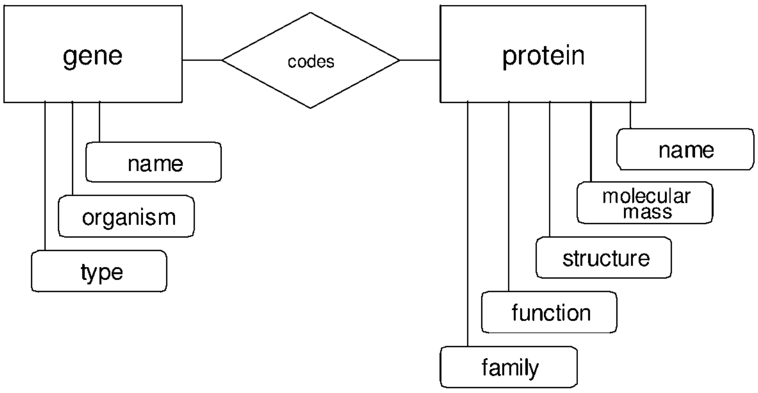


Figure 1 ER diagram.

The ER model is simple and intuitive and thus is easily understandable by database designers and users alike. The success of the ER model has lead to multiple extensions of the original model, e.g. to model generalization, specialization and aggregation, as published in numerous papers and books ([14,29,42] and references therein).

We illustrate how ER may model biological information in Figure 1. Gene and protein are defined as entities. Each entity is qualified by attributes represented in ovals linked to the entity they refer to. The relationship capturing the scientifically meaningful relationship a *gene codes for a protein* is represented with a connector between the two entities labeled codes. Although the model defines a *codes for* relationship between the entities *gene* and *protein*, an instance of the entity *gene* may encode no protein, one protein or many proteins.

Relationships may be characterized by one or several attributes in an ER diagram. For example, in a diagram aiming at representing literature references, two entities, i.e. author and journal, may be linked by a relationship "publish" with attributes date, volume, number, etc. These attributes characterize the relationship, not the entities. The ER model allows n-ary relationships that link n entities together, for any integer $n > 1$. For example, an alternative representation of the literature references would be to define a relationship "publish" between three entities: journal, date and author. In addition to these powerful basic representation mechanisms, the ER model has been extended to allow for the definition of a hierarchy of entity types (generalization hierarchy). Two entities, e.g. DNA and RNA, may be sub-entities (subtypes) of a common super-entity (supertype), e.g. sequence. When such a hierarchy is defined, all the attributes defined at the super-entity are propagated down the hierarchy to sub-entities. Like any modeling framework, there is no unique way to model the information. Often a representation is best suitable for a particular aim.

The ER model is a formidable tool for conceptual modeling. Not only it is easy to understand and use by both users and data providers, but it offers valuable advantages. First, it captures adequately the level of complexity of abstract objects that scientists need to represent. Then, unlike data models offered by data management systems, ER is not biased towards a particular data model and thus may be translated in the model of any computer system. This is especially important, as we will see that each data model typically twists the data with respect to its characteristics. Once an ER model is specified it is easy to translate it into available data models including relational, XML, etc. Many automatic and semiautomatic tools generate data structures from an ER diagram. Finally, as we will present in Section 5, the need for semantic integration raises the need for a formal conceptual model to represent the data organization. The ER model seems to be little used in the biological community. We find only 25 hits for "entity–relationship" retrieved from PubMed in September 2006. Still, its ease and the various technological support available make it a promising candidate for modeling conceptually biological data as needed for data representation and integration.

2.1.2 Unified Modeling Language

UML is a notation that helps documenting system specifications [10, 51, 72]. The language provides various diagrams that characterize the structure (class, composite, etc.), the behavior (use case, activity, state machine, etc.) and the interactions (sequence, communication, etc.) a computer system aims at achieving. The UML diagrams offer orthogonal abstractions or viewpoints of a soon-to-be designed computer system that overlap significantly. Although not all diagrams may be useful in every context, some of them appear to be beneficial to the scientists to represent the data organization of the data, but also to express the requirements of a system that will be designed to support bioinformatics tasks (see, in particular, the UML use case and activity diagrams).

The UML class diagram illustrates model elements such as classes and types, their content, and their relationships. In order to create and evolve a class diagram, the scientist needs to iteratively model the classes, responsibilities, associations, inheritance relationships and composition associations. We illustrate the expressive power of the UML class diagram with Figure 2 where seven classes are represented. It is worth noting that in Figure 2 all classes have the same weight and the diagram does not provide a data organization that is gene-centric or biased towards other characteristics. However, when translating this overall abstract data organization into the data structure related to a selected data model, the scientist may alter this uniform representation to adapt the data to the selected data model or to improve performance.

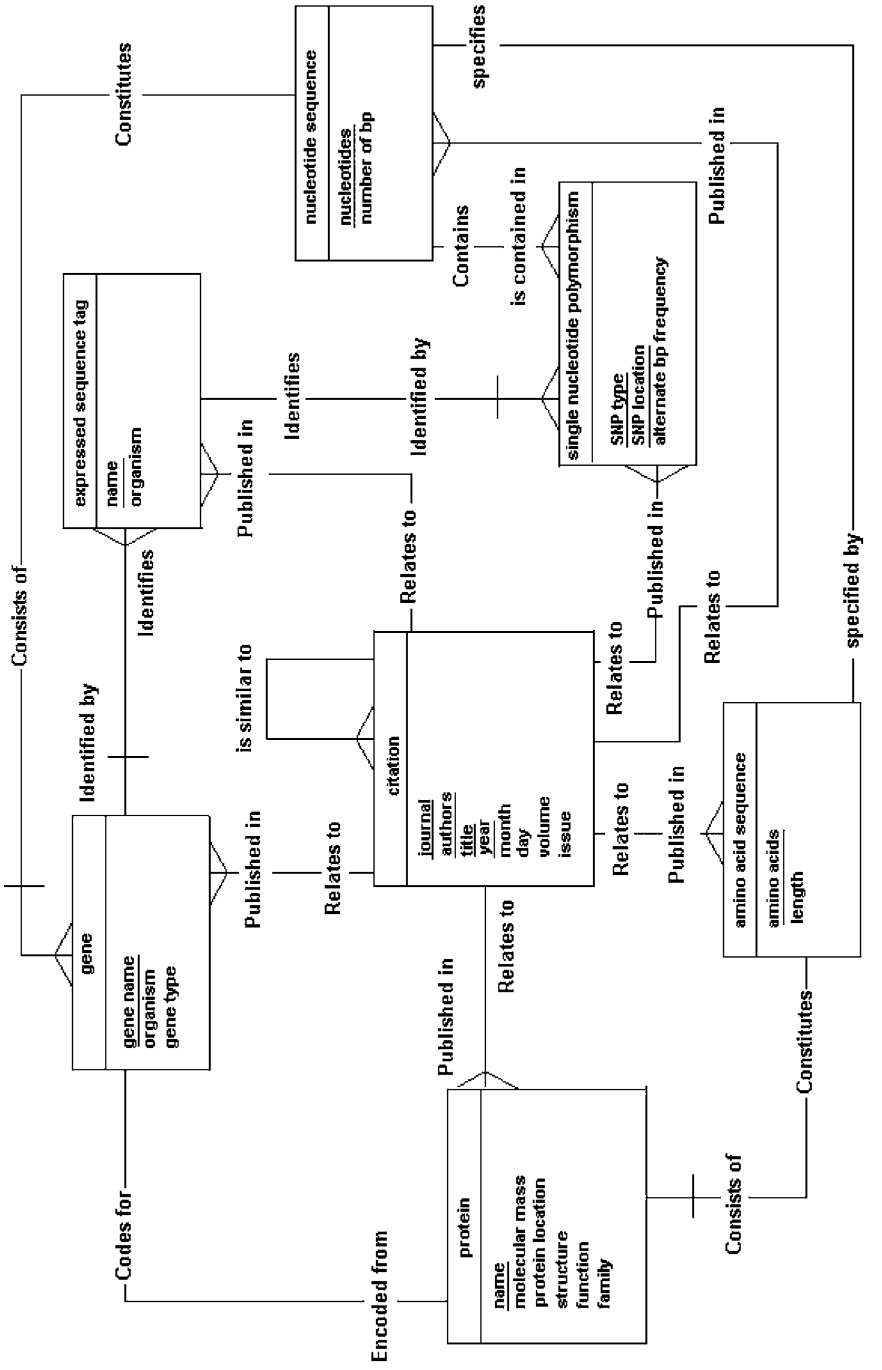


Figure 2 UML class diagram representing biological objects.

Although less critical for data integration, use cases are a powerful mechanism to characterize all interactions with the computer system which is being designed. Use cases are defined by the actors (system, users, other systems, organizations and groups) and the tasks (actions, interactions, and associations) that the system is expected to perform [47]. For example, a use case for searching the bibliographic references related to a particular genetic disorder could be characterized by an actor scientist and the following six tasks: (i) select the resource that provides information about genetic disorders, e.g. Online Mendelian Inheritance in Man (OMIM), (ii) enter the keyword in the form, (iii) click the search button, (iv) the search algorithm is executed, (v) the results are returned to the actor and (vi) the bibliographic references are extracted. In UML, a use case is characterized by a task linking two actors or systems. The UML use case diagram represents each use case by ovals labeled by the task it represents. In a use case, each system is represented by a rectangle that may, itself, contain internal use cases, i.e. the tasks performed by the system and the actors. The edges represent the interactions between the actors and the system, or between systems. Each edge links an actor (or a system) to a use case and a system. Tasks may exploit other tasks and labeled edges between tasks express these paths between actors (or systems) to systems.

The use of UML to model biological data and scientific activities is rather limited (127 entries in PubMed in September 2006), but it offers promising modeling for complex biological processes and systems [71] and efforts such as the development of BioUML to support systems biology are encouraging (http://www.biouml.org/model.shtml).

2.2 "Flat" Data Models

A flat data model uses a single mechanism (or container) to structure all information. For example, a textual document, as a string of characters, is a flat representation of information (*flat file*). The only level of structure provided by such a model is given by the named container (here the *document*). The relational model can be seen as a flat model as all information is contained in relations (*tables*). As complex as the conceptual representation of the information may be, the complete diagram can be represented as a single table (however, applying principles of database schema design and optimization [8] will generally lead to a multi-table organization). For example, the diagram shown in Figure 1 could be represented in a table with eight attributes – gene name, organism, gene type, protein name, molecular mass, structure, function and family. The relationship between two instances of, respectively, a gene and a protein will be implicitly expressed for the instances of a protein and gene on the same row. Such a row is also called a *tuple*. Alternative representations of information in the relational model rely on splitting the information among multiple tables, at the cost of maintenance (there exist various approaches to normalize schemata and minimize maintenance and problems with update anomalies due to redundancy). Resulting tables may be linked through foreign keys that express the relationship among scientific objects split over multiple tables, creating a complex graph of linked tables.

A relational database schema obtained from an ER diagram or from any other method, once used to structure data in a relational database system, may need some transformations to improve performance, integrity and maintainability. For example, an instance of a table Reference with attributes title, author, journal, date, and page number is likely to show a large amount of redundant data as a reference authored by five people will be stored with five rows each containing the same title, journal name, date, etc. In order to avoid these situations, relational schemata are normalized [8]. Normalized schemata gain in performance, but the resulting transformation often blurs the conceptual understanding of the data set. When integrating relational databases, approaches must extract (or reconstruct) from relational schema the logical organization of the data. Techniques such as database views, schema mapping (see Section 3.1) and semantic integration may facilitate the process.

2.3 Tree-structured Representations

Tree-like structures are often preferred by life scientists to represent scientific data. Maybe the legacy of classification combined with the overwhelming textual information stored in flat files (often nested) can explain this infatuation. Tree-like structures may be easily produced from an ER diagram by selecting an entity for root – all entities linked to the root entity via relationships become children of the root node, and so on. Illustrations of the process are given in [68].

The recent advent of XML and the development of nested relational data management system led to the development of the efficient management of data stored in tree-like structures. In XML, data are represented by elements and valued attributes in a tree. The XML data model was designed to accommodate the exchange of documents over the Internet. Markup languages such as the Standard Generalized Markup Language (SGML) and the HyperText Markup Language (HTML) were widely used to tag textual documents to facilitate their processing and web publishing, respectively. To overcome the limitations of HTML (where all tags are for display), XML was designed to extend HTML while being significantly simpler than SGML. Later, the XML data model was adapted for data management (see the XML schema and XML Query Language working groups at the World Wide Web Consortium, http://www.w3.org). Because of its legacy, the XML data model organizes data in a document (root of the XML tree), whose nodes are elements and attribute values, organized with respect to the order of their occurrence and depth of nesting in the document [1].

There are currently two categories of XML management systems: XML-enabled and native XML [17], although the borderline between them can sometimes be blurred. The first group includes traditional database systems extended to an XML interface for data collection and publication. However, the underlying representation is typically with tables. Examples of XML-enabled systems are Oracle10i (http://www.oracle.com) and SQL Server 2000 (http://www.microsoft.com). These systems were mostly designed to handle business tasks on the web. They have not yet been proven to be useful in scientific contexts. Native XML systems such as Tamino (a commercial XML management system from Software AG), ToX (an academic XML management system being developed at the University of Toronto, http://www.cs.toronto.edu/tox) and Galax (developed by Bell Laboratories of Lucent Technology, see http://www.db.bell-labs.com/galax) rely on data internally represented with XML trees, and should provide a flexibility interesting in the context of scientific data management. The implicit order of elements in a document, although very useful when handling textual data, affects significantly data storage and processing of XML enabled relational data management systems [61]. As biological data are for a large amount textual and deeply nested, XML native systems may be favored to access and store biological XML data.

The nested relational data model is an extension of the relational data model that allows attribute domains to be either atomic (as in the relational model) or valued (the value of the attribute is then a set of values, thus a relation). It can be a more efficient storage model than flat relational databases, as the nesting mechanism allows for implicit representation of relationships, and avoids redundancy of data over multiple tuples and tables. While being an extension of the relational data model, the nested relational data model possesses a query language as mathematically sound as the relational calculus [70]. The biological data management system BioKleisli [20, 26] uses the nested relational data model and calculus to manipulate internally scientific objects. Not only does the data model offer significant benefits, but the nested relational calculus appears to be a suitable language for expressing scientific protocols [34]. (See Ref. [49] for more information about those data models and their used in scientific data management.)

2.4 Graph Representations

There is a large variety of scientific data which are naturally represented as graphs. These data include all kinds of biological pathways and networks.

Metabolic pathways represent chemical reactions used for energy production, synthesis of proteins, carbohydrates, etc. Metabolic pathways can be represented by directed graphs, with nodes for reactions, inputs, outputs and catalysts. Genes, gene regulatory sequences and signaling proteins, which control the activation or suppression of gene expression, are structured in gene regulatory networks whose graph structure is similar to metabolic pathways. Note that these graphs are usually directed and cyclic. Cyclic directed graphs are typically difficult to query efficiently and also more difficult to process with functional programming methods, which have traditionally been developed for lists and trees. In contrast, protein interaction networks are undirected graphs. Graph operations such as *find a path between two nodes*, *find the shortest path between two nodes*, *transitive closure* (browsing through edges until all connected nodes are met) or *subgraph homomorphism* (exact match of graph structure – edges) express meaningful scientific queries.

Translating ER diagrams into graph-based data management systems is rather straightforward. However, there are not many systems supporting rich graph structure and graph query language such as needed in the life sciences. Research projects such as the Biopathways Graph Data Manager (BGDM) at Lawrence Berkeley National Laboratory (http://pueblo.lbl.gov/∼olken/graphdm/graphdm.htm#graphDataModel) are devoted to the design and development of a general purpose graph data management system to support biochemical pathways and protein interaction network databases for microbial organisms. A similar focus is the core of the MetaCyc effort at Stanford Research Institute (http://metacyc.org).

The primary purpose of Object-Oriented Databases (OODB) is to design a database that could easily model the world as classes of similar objects while being compatible with declarations of an object-oriented programming language such as Smalltalk or C++. In an OODB, the world is modeled in terms of a hierarchy of object classes, valued and abstract attributes, and collections [16]. A valued attribute assigns a value to an object, whereas an abstract attribute links two objects. The object-oriented data model seems similar to the ER conceptual model, but the differences between the two models are significant. The translations from an ER diagram to an OODB schema are straightforward as long as the ER diagram has been transformed so that all *n*-ary relationships, all attributes characterizing a relationship and all many-to-many relationships have been removed. Once the ER diagram is defined with only one-to-one or one-to-many binary relationships with no attribute, an OODB schema may be obtained by creating an object class for each entity, an abstract attribute for each relationship and a class hierarchy for each entity hierarchy. Unlike in the ER conceptual model, an object class can inherit from multiple classes.

The Object Protocol Model (OPM) is an object-oriented data model specifically designed to handle scientific data [19]. It allows for the definition of two different types of classes: object and protocol classes. As in traditional object-oriented data models, object classes in OPM are populated with identifiers of objects of the same kind. In contrast, protocols classes are designed to represent scientific experiments. An atomic protocol represents a process instance, with its input and output. Expanded protocols are composed of a complex network of connected atomic protocols. This expressive data model is used to develop the OPM multi-database system [45] and Web mediator [48]. The need to store scientific protocols together with the data they collect is still critical and is currently addressed with solutions based on workflows, as presented in Section 6.

2.5 Multi-dimensional Data Model

Scientific applications often generate large multi-dimensional arrays. These large data sets are often stored in files or spreadsheets that provide little or no meaningful structure. The multi-dimensional data model evolved from the processing of data contained in spreadsheets and used by business analysts in specialized multi-dimensional databases such as Express. The multi-dimensional data model consists of *facts* and *dimensions*. A fact can be regarded as an entity of an ER diagram and is represented graphically in the multi-dimensional data model as a data cube. Each dimension corresponds to a perspective under which facts can be analyzed, characterizing various measurement data related to the fact. Dimensions can be structured in hierarchies of levels for characterizing the modalities in which data can be grouped along dimensions. In a dimension, there may be more than one path along which to aggregate the data [82, 83]. Relational database systems often offer a multi-dimensional data representation by way of the star schema that structures the data in a cube. The star schema aims at structuring a set of mostly numeric measures and a set of dimensions that provide the context for those measures. A collection of dimensions uniquely defines each measure, while each dimension is specified by a set of attributes or a hierarchy of attributes. The star schema consists of a single fact table that contains the numeric or non-numeric measures and several dimension tables. The fact table is usually very large – the number of columns equals the number of dimensions it represents. In contrast, the dimension tables are usually significantly smaller, as they contain the non-numeric data associated with the attributes of the dimensions. While the star schema offers great performance and is intuitive for users to visualize, its limited multi-dimensional structure does not capture explicitly the hierarchy between different levels of aggregations [82].

Obviously, in the context of data integration it is critical to exploit the fact that data are organized in a data cube or in a star schema so that the multidimensional structure is understood when not made available to the user.

3 Data Integration

Once the data sources are selected, the first two challenges in data integration are to understand the data structures of the sources to be integrated and design a structure for the integrated data. When the analysis of the data structure has generated a framework for structuring the data and suitable mapping mechanisms for translating data from the remote sources to this integrated structure, data integration approaches may be chosen. Data integration approaches may be used to query heterogeneous resources or access data to populate a local data repository. Here, we present three traditional integration approaches: data warehouses, federations and mediations.

3.1 Scientific View of Data

There are multiple ways to represent data as presented in Section 2. Each data source has its own data organization, data model and data structure. When integrating these different data representation modes, integration methods often assume a *global schema*, thus a selected data organization and data model for representing integrated data. To provide the users with an integrated view of the data, any approach needs some formalism. This integrated view may be conceptual, if only the data organization is shown to the users, or structural, if the data are organized and structured with respect to a data model. Conceptual integration (illustrated in Figure 3a) maps integrated concepts to those of the integrated resources. The approaches presented in Section 5 typically use an ontology as a conceptual model and user interface for accessing data. Successful integration systems such as Transparent Access to Multiple Bioinformatics Information (TAMBIS) [76] and Knowledge-based Integration of Neuroscience Data (KIND) [57] were developed combining conceptual integration and mediation. Recent approaches aim at mapping not only scientific concepts, but also scientifically meaningful relationships [50].

In this section, we will focus on approaches based on structural integration (illustrated in Figure 3b). Although structural integration does not address conceptual integration, it does not mean that the organizations of data within the data sources to be integrated will correspond semantically, such that structural integration will be facilitated. Indeed, some data organizations maybe quite incompatible and require significant transformation in order to

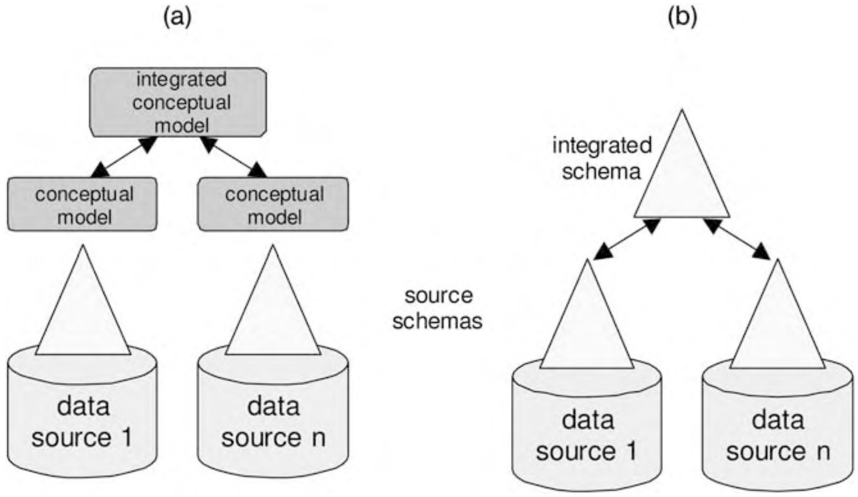


Figure 3 Conceptual integration (a) and structural integration (b). The triangles represent the data structure (or schema), while the ovals represent the conceptual diagram.

enable data integration. We list below some of the problems that the scientist may meet while integrating heterogeneous data sets.

- What is the integration problem?
  - Does the user already have an integrated data organization in mind? (Thus the customization of the integrated data set will be achieved with respect to users' requirements.)
  - Does the user wish to produce an integrated data organization that best meets the existing data organizations of the data sources to be integrated?
- What is the integrated data model?
  - Is the data model specified for the integrated data set?
  - Does it correspond to the underlying data sources to be integrated?
  - Does the user wish to select the data model that best meets the existing data models of the data sources to be integrated?
- What is the integrated data structure?
  - Is the schema specified for the integrated data set?
  - Does the user wish to produce an integrated schema that best map to the schemata of the integrated data sources?

Existing data integration approaches such as the ones presented in this section offer limited support to the problems listed above. Most of them assume that both the integrated schema and the mappings between the integrated schema and the source schemata are provided. They also offer limited support for maintaining the overall integrated architecture, e.g. with respect to changes at the source schema affecting the integrated schema or changes at the integrated schema affecting the mapping with source schemata. The development of *schema mapping* tools, that semi-automate the mapping between schemata, will improve this necessary support. Schema mapping tools typically assume that the integrated schema is known and map each of the source schemata to it by exploiting syntactic information (metadata) and semantic information (instance) in order to generate correspondences between structural components [28, 32, 84]. A general approach, applicable also in the context of data integration is *model management* [7]. In this approach, general schemata or "models" and mappings between them are considered as "first-class citizens" that can be stored, discovered, reused, transformed, etc., thus providing a useful data integration and transformation infrastructure. Recently, Clio, a schema mapping tool developed at the University of Toronto in collaboration with IBM, has been evaluated in the biological context of mapping the relational schema of GeneX (http://sourceforge.net/projects/genex) to the XML schema developed to exchange gene expression data GeneXML, formerly known as GEML [39].

Once the integrated schema and the mapping to (and from) all source schemata are specified, a *view* of the integrated data sources is defined. The notion of view was first introduced for relational database management systems to capture the customized transformation of an instance. A view is obtained by changing the data structure such that a certain category of users would only access the information they needed. Tables and attributes were often hidden to some users for security reasons. In the context of data integration, the data set shown to the user is an integrated view of data sources. The integrated schema is the structure of the view. A view may be *materialized* or *non materialized*. A view is materialized when an instance is created, accessing each integrated data source and loading the retrieved data into the new structure to create a new data source (thus materialized). When users query the view, they no longer access the integrated resources, but query the materialized instance. A view is non materialized when the view is used only as a querying interface and each time users query the view, queries are propagated to the integrated resources, retrieved data are integrated before being returned to the user.

3.2 Data Warehouse

A data warehouse is a collection of data integrated from multiple sources (databases, flat files, etc.) within a single system, usually a database. A data warehouse is a materialized approach as integrated data are downloaded – thus materialized – into the warehouse. Data from various sources usually need to be cleansed and reconciled before being integrated into the data warehouse. Maintenance is achieved at specified dates and within two updates the data accessed by the user in the data warehouse may be out of date.

The development of a data warehouse typically requires the selection of a data management system for storing the warehoused data and various resources from which data are collected. A warehouse schema is designed in order to structure the data in the system's format with respect to their chosen organization. The warehouse schema is a global schema as it structures data retrieved from various data sources and integrated in the warehouse. Once the warehouse schema is designed, queries or scripts are used to download data from the data sources, and various mechanisms are applied to curate and reconcile retrieved data that are reorganized with respect to the warehouse data organization and translated into the warehouse format. The resulting data source is independent from the resources from which it is generated. Therefore mechanisms are needed for its maintenance. In addition, the developers of the warehouse are also responsible for adding computational and analysis capabilities to the warehouse, capabilities that may be available at the integrated resource, but no longer usable.

The effort needed to develop and maintain a data warehouse often motivates the use of virtual integration frameworks such as a mediation approach (see Section 3.4) or partially virtual ones such as link-driven federations. Despite the additional workload, data warehouses are preferable in multiple settings occurring when data require significant curation or when users need privacy as well as for *data mining*. As data mining aims at discovering non-obvious relationships or trends in the data, data mining algorithms typically perform on data sets significantly less structured than in a traditionally data management system. The development of GeneExpress [58] illustrates the need for developing a data warehouse rather than using a virtual data integration system such as the OPM multi-database system.

When the data set includes statistics, a data warehouse exploiting the *data cube* may be a suitable solution for enhancing the ability to analyze large measurement data with respect to various orthogonal directions. On-line analytical processing (OLAP) relies on a data structure in terms of data cubes composed of data aggregated with respect to some parameters. In a nutshell, a data cube is a multi-dimensional table (see Section 2.5). This data model was introduced such as to overcome the limitations of the relational model which spreads the data over multiple two-dimensional tables and its query language [structured query language (SQL)] which does not support the expression of queries such as histograms, cross-tabulations, and roll-up and drill-down features [18].

3.3 Link-driven Federations

Although the terminology of database integration is not commonly agreed upon, we distinguish *federations* introduced in this section from *mediations* treated in the next section. Here, a federation is a partially virtual integration approach. It has a non-materialized feature as the data are not downloaded into any system and remain in the various systems that host them. A federation of databases is obtained by *linking* semi-autonomous, distributed databases. Each database has significant autonomy in the distribution while offering an interface to provide the user the capability to access integrated resources in a unified manner. In a federation, databases do not have total autonomy as they must maintain the links to the other members of the federation. These links constitute a materialized component of the integration and can be seen as a data set (typically a set of indices) that characterize the way data are integrated and need to be stored and maintained. Data providers may develop a federation to link the data source they develop. An example of such a federation for biological resources is Entrez developed at the National Center for Biotechnology Information (NCBI). For the user, the federation allows to query each of the integrated resources and conveniently navigate from one to the other. There is no global view provided to the user. A system such as the Sequence Retrieval System (SRS) was developed for maintaining a federation of multiple flat file data sources [30]. The system has since evolved to integrating flat files as well as structured databases and applications [31], providing an object-oriented integration schema.

3.4 Mediations

The concept of mediation was first introduced by Wiederhold to provide flexible modular solutions for the integration of large information systems with multiple knowledge domains [89, 90]. A mediation system (or *mediator*) integrates fully autonomous distributed heterogeneous data sources. In contrast to federations, mediations do not materialize any information. Instead they rely on *wrappers* to translate queries expressed with respect to the integration schema into queries to an integrated data source and translate the query results expressed in the source schema into data expressed with respect to the integrated schema. In contrast to the multi-database approach, mediators do not assume that all integrated sources will be relational databases.

Instead integrated resources can be various database systems (relation, object-relational, object, XML, etc.), flat files, etc. The major benefit of mediation systems is to always provide access to up-to-date data. An example of mediator is DiscoveryLink (also known as the DB2 Information Integrator and WebSphere) [38].

Although the approaches developed by the database community provide a powerful and efficient interface for accessing and transforming integrated biological data, they are often limited by the applications made available in order to analyze, simulate, visualize, etc., biological data. This limitation is due to the focus on data that ignores the integration of all resources, including applications. In the next section, we investigate how integration approaches developed for providing users access to all the useful applications they need (see Chapter 44) address the issues specific to data integration.

All integration approaches require expertise in order to comprehend the data organization, model and structure of each integrated resource. This expertise can be made available through valuable metadata (information about the data) for automatic processing and semantic integration as presented in Section 5.

4 Integrating Applications and Data

There are many technical solutions for integrating distributed resources into one application. These can be seen as the "plumbing" that joins resources together such that data can "flow" into one application. These technologies are explored in detail in Chapter 44, but a brief overview is provided here. These technologies for plumbing really address the lower levels heterogeneity: system and syntactic heterogeneity. At the system level, resources or applications run on different platforms, use different protocols and languages, and use a variety of call interfaces. At the syntactic level, the data are offered in a variety of formats. To overcome these hurdles it is necessary to "include" external resources into the local environment such that they appear to be present within the local application. Further, these external resources need to be "transformed" such that all resources appear to have the same syntax and a common behavior within the host application.

Most programming languages now have the necessary libraries, etc., to enable some sort of integration in an application setting. Java, for instance, has Java Database Connectivity (JDBC) by which an application can connect to a remote database and perform SQL queries upon that database. To overcome the syntactic heterogeneity between the "Standards" in SQL, there are layers such as Hibernate (http://www.hibernate.org) that will translate an incoming SQL query to that hosted by the target database. Programming languages also usually have facilities for importing web pages and consequently running services available via the Common Gateway Interface (CGI). All these facilities give partial access to distributed data resources, but they do not provide a complete or robust solution to the basic levels of heterogeneity that arise through distribution.

In this section we will briefly review the idea of middleware and some technical approaches to integration. By introducing some case studies from bioinformatics, we will soon see that "plumbing is not enough" [35a]. These technical approaches go as far as to bring all the data into one application, but those data themselves are still heterogeneous at the level of their structure and the values within this structure. This more difficult level of reconciliation will be dealt with in Section 5.

4.1 Middleware

To a programmer attempting to integrate bioinformatics resources within a single application, the fact that all the resources present different interfaces implies that a considerable amount of effort has to be expended. For example, consider the case when a Java program needs to access a remote database written in C++ and feeds some of the results to another program written in Perl. In order to do this the programmer must cope explicitly with the different languages in question – the C++ will be invoked in one way, Perl in another – and also deal with the distribution. The C++ program would likely be invoked differently if it were available on the local machine. As well as taking a large amount of effort, the resultant program is fragile. If it is decided to mirror the database locally, the Java program will need rewriting. If the Perl program is ported to C++, again the Java needs rewriting.

One solution to this problem is to use some *middleware* technology. As the name suggests, this adds a middle architectural layer that abstracts away from the different languages, systems and locations. It does what its name suggests – it sits in the middle between the application layer and the underlying resource layer. Following the example above, instead of writing code in Java to invoke the C++ database directly, both will be *wrapped* with the middleware technology. This technology then has the task of managing the communication between these wrappers. While this seems overly complex, it actually simplifies many issues. The Java programmer no longer has to worry whether the C++ database is local or remote nor does it matter that interaction is needed with both C++ and Perl. Therefore, middleware technologies offer an attractive solution to overcoming system and syntactic heterogeneity in a distributed setting.

4.2 CORBA

CORBA is one middleware solution to the problems of integration. It arose in the 1990s, and is now a mature industry standard and was widely proposed as a solution to the problem of integration [85], especially in bioinformatics [78, 85].

CORBA attempts to present a common view of the world by presenting it from an object modeling perspective. To continue the example introduced earlier, to the Java program both the C++ database and the Perl program would appear to be Java objects. Interaction with these objects would be identical to interaction with any other Java object. Similarly, on the C++ side, queries to the database would appear to be coming from a local C++ object rather than from remote Java.

In order to enable this technology, the target resources can be described in a common language. This common language can then be compiled automatically into the programming language of choice, which then enables these target resources to appear as if they were part of the local host application. A core feature of the CORBA specification is this language – the Interface Definition Language (IDL) [78]. This language is used to describe what operations, including return types and arguments taken, target resources perform, and it can be used by CORBA compliant tools to generate code for both providing access to the services and the means for the services to be accessed. One low-level task performed by CORBA is to define how much memory is used for primitive types such as real and integer numbers. Typically on a PC an integer uses 2 bytes of memory and UNIX machine 4 bytes of memory. This system-level heterogeneity needs to be smoothed away, otherwise numbers too big for a platform will cause errors at the application level.

Once described in IDL, a platform-specific compiler takes this code, and generates skeleton and stub code for each class, attributes and operations upon that class. The skeleton code includes all the code needed to support a server-side application for hosting the remote resource. There is code generated for the client side; it includes all code necessary for calling the remote object and simply makes the remote objects appear as if they were located "in" the host application. These requests from the application are all processed via an Object Request Broker (ORB) which, as its name suggests, brokers requests between objects. By this seemingly heavy and complex set of procedures, it is possible to build robust, integrated applications including distributed resources running on different platforms, using different languages.

The European Bioinformatics Institute (EBI), in particular, invested a great effort in providing CORBA solutions for many services, including EMBL (http://corba.industry.ebi.ac.uk) [40, 67, 69]. The Object Management Group (OMG) formed the Life sciences Research Group (LSR) that has developed several standards for services including bibliography and sequence resources (http://www.omg.org/lsr). However, the uptake of CORBA by the community has not been widespread. The main reasons for this have been the perception that CORBA is too heavyweight a mechanism – the large effort required to develop the standards seen as necessary by the Object Management Group (OMG) and the implementations themselves obstructed development. Many of the early ORBs were expensive and focused on enterprise-level computing, which did not fit well with the bioinformatics cottage industry. In addition, many ORBs themselves did not actually inter-operate. Finally, CORBA seemed to be plagued by continual problems with tunneling through firewalls, defeating the promise of location independence.

4.3 Web Services

Web services [25] take the same basic approach to distribution as CORBA, but with several significant differences. At heart, both take a description of a service being offered, and produce code for developing clients and servers. Web services takes the view that distributed tools and data are offered as "services" to applications that wish to use them. Any such service-orientated architecture has the following components:

- A standard communication protocol between services and host applications.
- A uniform data representation and exchange mechanism.
- A standard language for describing the service's attributes and operations.
- A mechanism for registering and discovering web services.

All these have their counterparts in CORBA, but web services take advantage of developments in web technology not apparent during CORBA's earlier development. These are the standard Internet protocol HyperText Transfer Protocol (HTTP) and XML. The approach also tries to make delivering a Web Service as lightweight as possible.

The Simple Object Access Protocol (SOAP) is the channel used for communication between a web services provider application and a client application [25] (http://www.w3.org/TR/soap). SOAP re-uses HTTP for transporting messages. Messages are passed between services using XML documents. The structure for SOAP message includes an *envelope* and a *body*. The body itself describes a message to a service, for instance, a call to a particular operation or communicates failure. The envelope gives the metadata necessary for this invocation.

As CORBA's IDL is used to describe services, web services use another XML document type to describe services – Web Services Description Language

(WSDL). Just as with CORBA, these descriptions are compiled to generate client and server code. These can be for a variety of programming languages on a variety of platforms. Client and server, once deployed, are ready to pass SOAP messages between one another. Finally, just like CORBA's naming and trading services, web services need to be discovered for use. WSDL documents can be placed in a registry based on the UDDI framework and these registries can be searched to retrieve WSDL descriptions of interest. A user would then compile to generate a client and use information from the WSDL description to locate and use the service.

Web services take a different technology approach to that of CORBA. While the latter uses a remote object approach which provides the ability for passing around data and subsequent fine grained client–server interaction with that data, web services use a "document-based" paradigm. Here, potentially complex structured data is passed between services in bulk. The hope is that instead of a series of fine-grained interactions between client and server, fewer coarser, but richer, interactions will happen – something of clear benefit when faced with any serious problems of network latency or failure.

Web services have already seen a much higher uptake than did CORBA. Taverna, a bioinformatics workflow editor [63], can currently access over 1000 web services, with many of these being third-party services. Stein [75] sees web services as the technology that has the possibility of uniting the fragmentary bioinformatics world. Where CORBA took the resources of a large institution to deploy, web services can easily be provided by a single user. This is why we already see many services provided by many institutions.

4.4 P2P

P2P architecture models a network of autonomous systems as a graph of nodes called peers (Figure 4b). Unlike the client/server architecture (Figure 4a) used in distributed computing models where some computers are dedicated to serving others, in the P2P architecture all participants can be alternatively clients or servers in the network. In a P2P architecture participants rely on one another for service as each peer is expected to offer a service, typically sharing its CPU with other peers.

Many P2P networks have been developed and the evaluation of such architectures has been published in multiple papers (a useful bibliography maintained until 2003 is available at http://www.cs.toronto.edu/db/hyperion/bibliography.html). Often P2P systems have been developed for sharing data over the web. For instance, Napster is a P2P network devoted to sharing music. Still, such systems can also be used in domain-specific contexts such as bioinformatics where scientists cannot only share their CPUs, but also

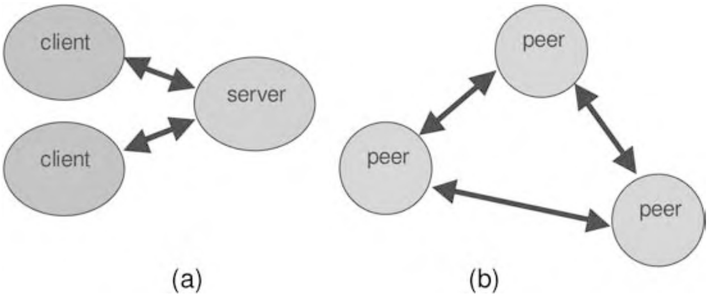


Figure 4 (a) Client–server. (b) P2P

share integrated resources, access to databases or applications alike. Such communities are implemented as *grids*.

4.5 Grid

It is easy to see how the Grid paradigm (see Chapter 44) of sharing resources fits into the world of bioinformatics: a sophisticated, complex bioinformatics *in silico* experiment may involve people, many forms of data, instruments, etc., and all of these could share resources in a Grid [36].

High-throughput biology is forcing bioinformaticians to adopt distributed computing solutions, such as Grid, within bioinformatics – it simply has to happen. In addition, many bioinformatics resources have realized the need to overcome the problems of semantic integration (see Section 5). From the technical side, web services and Grid computing are coming together to form a potentially lightweight, transparent access to the problems of system heterogeneity and distribution. On a more prosaic level, funding bodies see the Grid as a potentially high-impact area and funds have been available for large scale projects using these technologies. Finally, much of the Grid and web world follows an open-source agenda, suggesting a larger uptake of the technology. Despite the continuing problems described below, the cases described in Chapter 44 suggest that some aspects of integration will become much easier.

5 Semantic Integration

A challenge in data integration is to properly understand the data provided by a resource such that they are meaningfully integrated. To address this challenge, there is a need to integrate or at least make use of information and knowledge rather than data only, and exploit all the information related to the way the resource makes these data available. As illustrated in Figure 5, we

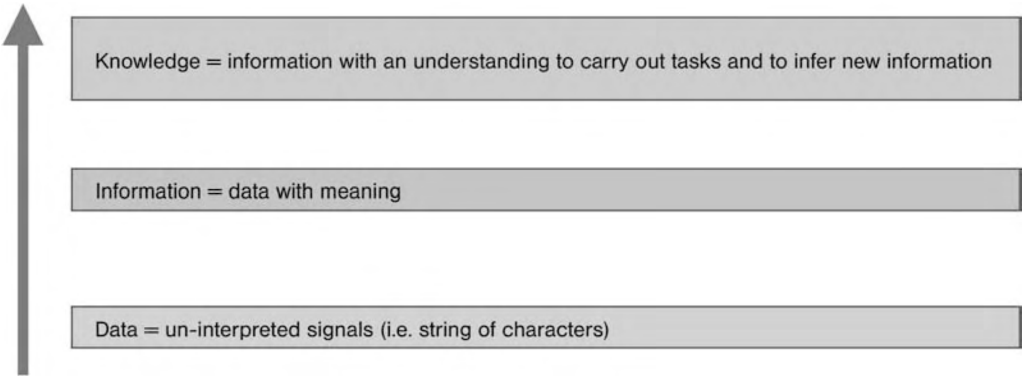

Knowledge = information with an understanding to carry out tasks and to infer new information
Information = data with meaning
Data = un-interpreted signals (i.e. string of characters)

Figure 5 From data to knowledge.

refer to *data instance* as a string of characters carrying no meaning, because no context or metadata are given. In contrast, we speak of *information* and *knowledge* when additional metadata and context information provide some level of meaning or "understanding" of the data. For example, *tp53* without further context is only a data string of four characters, while *gene TP53* can be considered information, with "gene" providing a minimal context to help a human or a system "interpret" the subsequent data string.

Metadata are data about data and thus are used to describe the content, context, and "meaning" of a resource. Metadata add value to the data so that together they define information. For example, a data source using a relational table called "gene" with an attribute "symbol" provides the structural terms gene and symbol as metadata. When extracting the value TP53 from the resource, this is only a string of characters, but with the knowledge that TP53 is the value of attribute symbol of table gene, we obtain information: *gene symbol TP53*. (In a data integration context, the context/metadata string "gene" might help the system determine which tables to access to find related information. Moreover, "gene" might be a term from a controlled vocabulary or a concept from an ontology, thus providing additional context information about "TP53".)

A complementary challenge in data integration is to provide users with a meaningful view of the integrated data set. While data may be internally represented and stored in a complex data structure poorly reflecting the users' understanding of the data, it is critical that the users are provided a meaningful view of the data set. Such a view is closer to the conceptual view as expressed with ER. In this section we present some approaches that exploit semantics to either provide useful metadata to facilitate data integration or to specify the overall organization of integrated data to users.

5.1 Identifying Objects

Each time a scientific object is stored in a data repository (e.g. a database) it is typically assigned an identifier. The problem is that as the number of resources providing information about each scientific concept (e.g. protein) increases, the number of identifiers for each instance of the concept multiplies. Although two data sources may provide information about the same concept, the challenge for integrating the data from the two resources relies on the mapping of two different identifiers. Sometimes, a scientific entity has not only as many identifiers as the number of data sources that contain information about it, but it does not possess a name upon which the entire community agrees. These combined discrepancies make it very difficult for humans and systems alike to refer to entities without ambiguity. To overcome the lack of naming process, the scientific community has adopted identifiers to name, or to refer to in an unambiguous way, the scientific objects. Examples of this adoption mechanism can be found with literature references typically identified for PubMed identifiers and sequence accession numbers from GenBank. Once these identifiers are adopted by the community, they become new attributes for most of the resources providing information about the scientific concept to which they refer. (In general, while referring to the commonly accepted identifier, the data provider still keeps its redundant internal identification process.) This avalanche of different ways of identifying an instance may be overcome by various resources that aim at providing for each entry the mapping between many existing identifiers in various data sources. GeneCards [66] and Genew [86], both databases of human genes, can be seen as playing this role. The effort to resolve conflicts in identification in life sciences led the Object Management Group (OMG) together with the Interoperable Informatics Infrastructure Consortium (I3C) in conjunction with vendor members including Sun Microsystems and IBM to design an identifying framework called Life Science Identifier (LSID) that locates scientific entries within the resource that hosts them (http://www.omg.org/technology/documents/formal/life_sciences.htm). These identifiers play a role similar to URLs to locate web pages. LSID uses a uniform resource name (URN) that contains five parameters that uniquely identify the data of interest [73]. An example of an LSID is: urn:lsid:ncbi.nlm.nig.gov:GenBank:T48601:2. The five parameters are: a mandatory preface for LSID data (urn:lsid), the Internet domain of the organization that assigned the LSID to the data (ncbi.nlm.nig.gov), the name of the data source (GenBank), and the last two parameters are the name or identifier as defined by the data source authority (T48601) and version number (2) of the scientific entry. The use of identification frameworks such as LSID offers numerous benefits. It clarifies the identification of scientific objects with respect to each data provider. Instead of refereeing to an often meaningless string (e.g. T48601) whose interpretation may be ambiguous, identifiers such as LSIDs carry enough context information to avoid misinterpretations and ambiguity. Obviously, such identifiers will be valuable to express the multiple links between entries among heterogeneous distributed resources. Another advantage of LSID is that when automatically processed, systems cannot only access the data source (the URL is included in the URN), but also may link that resource with a known data entry format (e.g. a XML DTD) in order to process automatically the entry once retrieved [73]. The involvement of database management systems vendors in the design of this identification format illustrates the need for meaningful identification in the life science. However, using such a common format to identify scientific objects does not resolve the problem of unique identification. For example, the two distinct LSIDs urn:lsid:au.expasy.org:SwissProt:U64442:1 and urn:lsid:www.gene.ucl.ac.uk:HUGO:APC both refer to the same scientific object hosted in two different data sources, respectively UniProtKB and Genew. Resources aiming at maintaining identification mappings between resources combined with a specific and meaningful identifier such as offered by LSID may be the solution to the current challenge of identification and location of scientific data. Although LSID will not solve the problem of redundant identification of scientific objects, it will facilitate the unambiguous mapping of equivalent identifiers for better data integration. Various viewpoints or contexts may affect the meaning of scientific object identity and, thus, life science identifiers should have the ability to match the different senses of "sameness" that pervade the life sciences [68].

5.2 Representing Metadata

Data are only as useful as the metadata that accompany them. Metadata, the data about data, inform the user, whether computer or human, about how to interpret those data. We use metadata at every stage of any bioinformatics representation or analysis. Take, for instance, a UniProt/Swiss-Prot record. The real data, the sequence itself, is only a small proportion of the record. Most of the record is taken up with management information (accession numbers, identifiers, change dates, etc.), descriptive metadata (gene names, species, organelles, keywords, meaningful names, etc.) and then broader and more detailed knowledge about the sequence (individual features, comments on disease and function, etc., cross-links to other databases, bibliographic references, etc.). All this is metadata about the sequence. Of course, what is viewed as metadata depends on where the user is standing – we are well acquainted to using metadata as data. There is data about data about data, etc.

How should all these metadata be represented so that they are most useful? Data without metadata are almost useless, but the usefulness of those metadata very much depends on their representation. Metadata can be captured in a whole spectrum of presentations: natural language (glossaries), structured vocabularies, frame-based representations, formal systems, through to logic representations. This spectrum covers the range from that only really interpretable by humans to that interpretable by computers and, with some help, by humans.

Bioinformatics metadata fall into two categories – the schema that structures the data and the values that describe the data. For most of its history, the bioinformatics community has used natural language or stylized natural language to describe its data. In the early days, with little data, this was easy, as only humans used those metadata. With the growth of volumes of data and creeping heterogeneity at the semantic level, this simple approach became untenable.

Controlled vocabularies were the next step. Simple keywords were used to describe database entries, such as Swiss-Prot keywords, and eventually these keywords were defined. We now see the growing use of vocabularies generated from ontologies to provide semantically rich, consistent metadata. Similarly, structures for data have moved from proprietary ASCII encodings to increased used of relational schemata, XML and also ontologies to provide the structure in which values are placed. Various efforts to integrate data demonstrated the role of generic schema and structured vocabularies, which will be described below.

The Resource Description Framework (http://www.w3.org/TR/rdf-concepts) is one of a range of technologies that aims to support the Semantic Web. This is the vision for the next stage of the development of the web, where it will move from an artifact only really interpretable by humans to one interpretable by both humans and computers [6]. This move is predicated upon the use of metadata that will describe both the content and services available on the web. The Resource Description Framework (RDF) is the means by which this metadata will be provided.

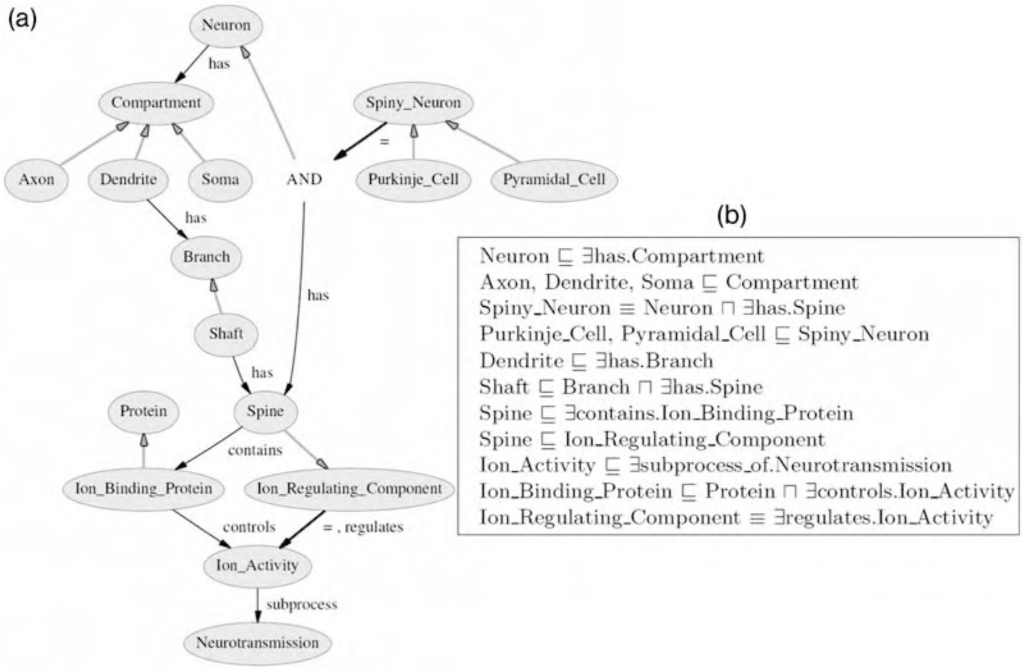
RDF uses a system of triples – a subject, a predicate and, finally, an object – to describe resources. For example, a resource or subject might be a UniProt/Swiss-Prot record, a predicate would be "has accession number" and the object would be "P21598". The object can be another resource, rather than only a value, so we could describe that sequence A is similar to sequence B. Each of these elements has its own URN (see above) to identify it. The mass of triples forms a graph describing resources. What is important is that the RDF triples have a simple, but formal semantics. This means that they are easily exported from data sources into this simple format and then interpretable by computers.

Graphs can be aggregated, stored in RDF stores [13, 91] and queried with SPRQL – an RDF query language (see http://www.w3.org/TR/2005/WD-rdf-sparql-query-20050721). We are beginning to see more data in bioinformatics described using RDF – UniProt/Swiss-Prot (http://www.isb-sib.ch/∼ejain/ rdf) and Affimetrix (http://www.affymetrix.com/community/publications/affymetrix/tmsplice/index.affx) provide a great deal of genomic data described using RDF. YeastHub [22] is an example of integration using RDF technology. Here, a series of third-party yeast resources are simply transformed into RDF and automatically integrated by virtue of RDF's simple data model.

Whatever the representation for these metadata is, they have to represent a shared understanding for their domain. As already described, one of the main barriers to integration is the heterogeneity in the semantics of the data being integrated. Integration becomes much easier if the semantics of those data are described in a form that captures the understanding of their community [79]. The most conspicuous example of this is the Gene Ontology (GO)'s use for describing the major attributes of molecular functionality across bioinformatics [81]. Similarly, we see generic representation of schema for certain domains, such as pathways in BioCyc [46], protein families [92] and in XML schema for many domains [44,88]. One technique for describing both the structure and values for a domain are ontologies and these will be described in the following section.

5.3 Ontologies and Data Integration

In the following, we briefly introduce ontologies and describe the role they play in the context of data integration; for a detailed exposition of ontologies in molecular biology see Chapter 29. In computer science, *ontologies* are artifacts used to (partially) formalize a domain of interest by identifying and uniquely naming relevant concepts and the relationships between them. In the broadest sense of the word, controlled vocabularies, thesauri and taxonomies (classification hierarchies), even database schemata are sometimes called ontologies. In the narrower sense, ontologies are often defined to be "explicit and formal specifications of a conceptualization" [37]. Description logics (DLs) are decidable fragments of First Order (FO) logic that are often used to define ontologies. [This means that there are algorithms for answering certain questions, e.g. is concept C subsumed by concept D (an implication problem), is a concept definition satisfiable or consistent with another concept, etc.] The vocabulary of a DL ontology consists of a set of *concept* (or *class*) names and a set of *roles* or *relationships* which are used to interrelate concepts. Specifically, an ontology consists of a set of *axioms* interrelating concept names



$\text{Neuron} \sqsubseteq \exists\text{has.Compartment}$
$\text{Axon, Dendrite, Soma} \sqsubseteq \text{Compartment}$
$\text{Spiny\_Neuron} \equiv \text{Neuron} \sqcap \exists\text{has.Spine}$
$\text{Purkinje\_Cell, Pyramidal\_Cell} \sqsubseteq \text{Spiny\_Neuron}$
$\text{Dendrite} \sqsubseteq \exists\text{has.Branch}$
$\text{Shaft} \sqsubseteq \text{Branch} \sqcap \exists\text{has.Spine}$
$\text{Spine} \sqsubseteq \exists\text{contains.Ion\_Binding\_Protein}$
$\text{Spine} \sqsubseteq \text{Ion\_Regulating\_Component}$
$\text{Ion\_Activity} \sqsubseteq \exists\text{subprocess\_of.Neurotransmission}$
$\text{Ion\_Binding\_Protein} \sqsubseteq \text{Protein} \sqcap \exists\text{controls.Ion\_Activity}$
$\text{Ion\_Regulating\_Component} \equiv \exists\text{regulates.Ion\_Activity}$

Figure 6 Ontology in graph form (a) and formalized in DL (b).

and relationships, thereby constraining the set of possible interpretations for these symbols.

In the context of databases, ontologies may be used to describe the information pertaining to a particular scientific object, thus providing the foundation for data characterization and schema definition. Such usage is presented in detail in Chapter 29. Another use of ontologies consists of the description of how various scientific objects interact. With an ontology, one may define an integrated conceptual model (see Figure 3 in Section 3.1). This second usage mentioned in Chapter 29 is described in detail in this section.

5.3.1 **Example**

Figure 6 depicts an ontology in graph form (left) with oval *nodes* representing concepts of interest (here from a neuroscience domain: Neuron, Axon, Dendrite, etc.) and *labeled directed edges* expressing binary relationships between concepts. For example, has(Neuron, Compartment) means that every Neuron has some Compartment(s); the upward unlabeled arrow means "is a", e.g. isa(Soma, Compartment), i.e. soma is a compartment. In the center-right of Figure 6 the ontology axioms are given as formulas in DL (logical constraints). For example, the first line in the DL ontology is syntactic sugar for the FO logic formula:

$$\forall x\ [\text{Neuron}(x) \rightarrow \exists\, y\ \text{has}(x, y) \wedge\ \text{Compartment}(y)],$$

which states that if x is a neuron, then there exists some y which is part of x (i.e. "x has y") and which is a compartment.

5.3.2 **From Information to Reasoning**

Domain knowledge expressed in DL can be seen as a kind of "formal metadata", potentially capturing more semantics than their informal cousins, i.e. controlled vocabularies, simple taxonomies and thesauri:

- As mentioned earlier, *controlled vocabularies* already provide a simple mechanism for uniquely and consistently naming object classes of interest. However, there is little additional functionality – beyond consistent naming, which already is a significant achievement in practice – that is offered by controlled vocabularies.

- A *taxonomy* provides a classification mechanism, typically expressed as a subclass–superclass relationship. In biology, a taxonomical unit (*taxon*) is a named group of organisms. Taxa are organized into a hierarchical scheme. Traditionally, scientific names have been directly used as taxa, but this leads to ambiguities due to changes in the naming schemes and the meaning of a name over time [43].

- A *thesaurus*, creates relationships between terms of interest using special relationships such as "broader_than(x,y)" and "synonym(u,v)", stating that the term/concept x is broader than y, and that u and v can be used interchangeably. It is easy to see that an information retrieval system or a database system can exploit such information when answering user queries. In particular, when a user asks for information on x, then synonyms and subconcepts of x (and sometimes even superconcepts) can be listed. (The same is true for taxonomies.)

Ontologies can further refine the domain knowledge capture by providing additional constraints that the other formalisms cannot provide. A basic idea of ontologies, especially when formalized as DL axioms, is to define (new) concepts in terms of other (given) concepts. For example, in Figure 6, we define the constraint expressing that for something to be a *Spiny_Neuron*, it is necessary and sufficient to be a neuron and to have spines; in DL:

$$\mathit{Spiny\_Neuron} \equiv \mathit{Neuron} \cap \exists \mathit{has.Spine}.$$

Translated into FO logic this statement becomes:

$$\forall x\ [\mathit{Spiny\_Neuron}\ (x) \leftrightarrow [\mathit{Neuron}(x) \wedge \exists y\ \mathit{has}(x, y) \wedge \mathit{Spine}(y)]].$$

From this logic axiom it follows the following statement (among others): spiny neurons are a subclass of neurons.

In order to compare the modeling capabilities of the different formalisms, consider, for example, a controlled vocabulary that includes the terms *Monocyte*, *Leukocyte* and *Macrophage*. The use of such a vocabulary limits (in a positive sense) the number of "allowed terms", e.g. when annotating a microscopic image. A thesaurus or taxonomy based on this vocabulary can add more information, e.g. by saying "isa(*Monocyte*, *Leukocyte*)" or "synonym(*Monocyte*, *Macrophage*)". In this way, when running a database search for "*Leukocyte*" or "*Macrophage*" related information, a system can take the information captured by the thesaurus or taxonomy to expand the search in a way that guarantees that all relevant data is returned. Using an ontology, even more information can be specified. For example, we might want to define that macrophages are those monocytes that occur in tissue:

$$Macrophage \equiv Monocyte \cap \exists occurs\_in.Tissue.$$

Additional information could be captured in an ontology, e.g. on the process that has a monocyte move into the tissue.

5.3.3 Biological Ontologies

Over the years, life science has generated multiple vocabularies and more complex conceptual models to organize, represent, and share scientific knowledge. The Medical Subject Heading (MeSH) published by the National Library of Medicine is a controlled vocabulary used for indexing articles of Medline/PubMed (the complete description of MeSH is available at http://www.nlm.nih.gov/mesh/meshhome.html). The 22 568 MeSH descriptors include broad headings (e.g. anatomy) and specific ones (e.g. ankle) arranged in both an alphabetic and a hierarchical structure, and linked with thousands of cross-references that assist in finding the most appropriate MeSH Heading. An example of a large meta-thesaurus (i.e. combing information from multiple thesauri) is the Unified Medical Language System (UMLS) from the National Library of Medicine (http://www.nlm.nih.gov/research/umls/umlsmain.html). A prominent biological ontology is GO. The GO project is a collaborative effort to address the need for consistent descriptions of gene products in different databases (http://www.geneontology.org). It aims at capturing information about molecular functions, biological processes and cellular components. Additional biological ontologies include RiboWeb which structures the ribosome as well as supplementary functional data (http://smi-web.stanford.edu/projects/helix/riboweb.html) and the Fungal Anatomy Ontology (FAO) (http://www.yeastgenome.org/fungi/fungal _anatomy_ ontology/index.html) which is a controlled vocabulary that describes the anatomy of fungi and other microbes.

Conceptual models in all their forms, from the simpler vocabularies to the most complex and expressive logics, are increasingly used for biological data integration. They are used to provide meaningful annotations to biological data to improve data access or data sharing and integration through textual search engines. Examples of such uses include the interface provided by NCBI to search PubMed visa MeSH terms (http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db =mesh) and the annotations of EBI. UniProt UniProtKB/Swiss-Prot has joined the GO Consortium and has adopted its standard vocabulary to characterize the activities of proteins (http://www.ebi.ac.uk/GOA). Ontologies may be used to provide users with a meaningful data representation of integrated data as explained in Section 3.1 and illustrated in Figure 3(a). A precursor was TAMBIS developed at the University of Manchester (see http://imgproj.cs.man.ac.uk/tambis) [79]. More recently, KIND, like TAMBIS, has combined mediation and ontology approaches to integrate heterogeneous autonomous data sources [55]. KIND was developed at the Super Computer Center of San Diego (http://www.npaci.edu/DICE/Neuro). Current approaches aim at mapping not only scientific concepts, but also scientifically meaningful relationships [50]. Conceptual models may also be used to provide scientists valuable support when exploring the wide scope of available resources to express their scientific protocols. Path-based guiding systems use conceptual models to annotate the graph of existing biological resources (databases and applications) and return paths on physical resources to evaluate scientific protocols expressed against the conceptual model. BioGuide [24] (http://www.lri.fr/∼cohen/bioguide/bioguide.html) and BioNavigation [50] (http://bioinformatics.eas.asu.edu/BioNavigation.html) are examples of such systems.

5.3.4 **Ontologies and Data Integration**

Figure 7 illustrates a number of uses of ontologies and articulations (mappings between ontologies) in a data integration setting [11, 55]. Figure 7(a) shows an architecture which includes, in addition to standard data access services (e.g. to run SQL queries on a database), also "registration services" to register schemata of data sources with one or more ontologies. Based on these source-to-ontology mappings, a semantic mediation service can support interlinking and grouping of data based on conceptual level information (Figure 7b). Sometimes more than one ontology needs to be considered. In this case, *articulations*, i.e., mapping between ontologies, can be employed, essentially creating a "merged ontology".

When resources already provide a description of their data with ontologies, these ontologies need to be integrated to integrate the underlying data sets. When several ontology editors such as Protégé [60] (a free, open-source ontology editor and knowledge-base framework developed at Stanford and

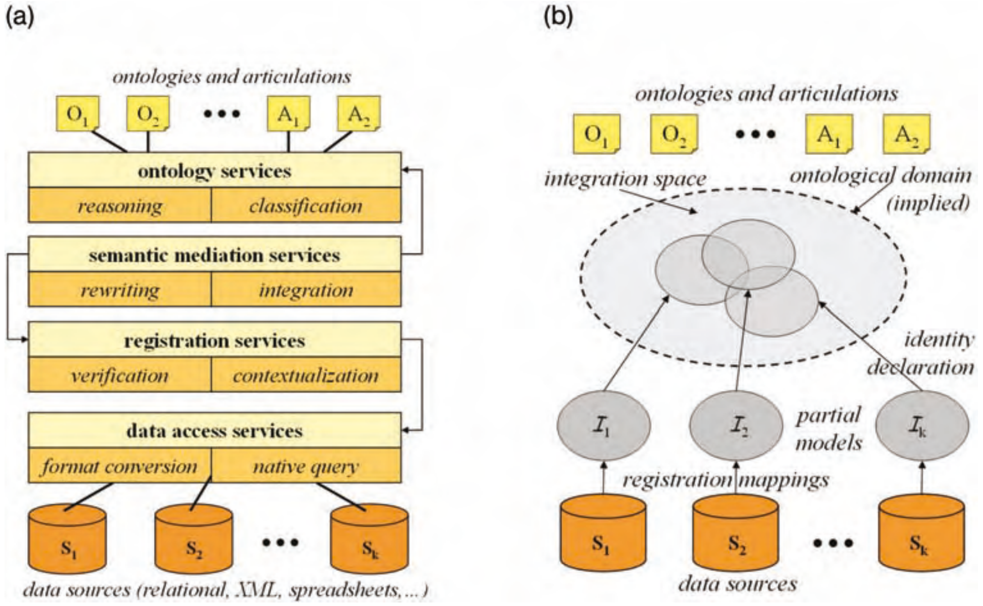


Figure 7 Semantics-enhanced data integration architecture (a) and integration mappings from the data domain to the ontological domain (b) (see Ref. [11]).

available at http://protege.stanford.edu), Chimaera [59] (developed at Stanford and is available at http://www.ksl.stanford.edu/software/chimaera) or OilEd [4] (an ontology editor developed at the University of Manchester and available at http://oiled.man.ac.uk) allow scientists to define and manage their ontologies, systems such as PROMPT [62] and ONIONS [23, 35, 65] (http://www.loa-cnr.it/Ontologies.html) allow the integration (merging) of ontologies, thus facilitating the data integration process.

5.4 Semantic Web

The Semantic Web [6] is a project for developing technologies and standards that capture some of the semantics of documents, particularly web documents. The vision is that Semantic Web technologies will enhance conventional searching, browsing and integration techniques by taking advantage of machine-processable annotations of documents and data. The Semantic Web can be viewed as a set of standards including XML, RDF and OWL (Web Ontology Language). A description of the features of the language OWL is available at http://www.w3.org/TR/owl-features. In general, the most up-to-date and complete information about the Semantic Web activity, including documents and announcements regarding related implementation, is available at the working group web site hosted at the World Wide Web Consortium (http://www.w3.org/2001/sw). XML, introduced in Section 2.3, is a format developed to exchange data over the web. RDF introduced in Section 5.2 is the framework developed to describe resources on the Web. OWL is a family of web ontology languages, i.e. OWL-Lite, OWL-DL and OWL-Full. OWL-Lite supports classification hierarchies and simple constraints and is easy to process, whereas OWL-DL corresponds to a DL, i.e. a decidable fragment of FO logic. OWL-DL is more expressive than OWL-Lite, while retaining computational completeness, i.e. questions about whether an OWL-DL formula is implied by a set of OWL-DL axioms (an ontology) are still decidable. In contrast, OWL-Full offers maximum expressiveness, but is no longer decidable. A logic is decidable if there exists an algorithm such that for every formula the algorithm is capable of deciding in finitely many steps whether the formula is valid or not. A decidable logic allows a mechanism to answer questions expressed in the logic, thus validating or not any given statement. As biological ontologies aim at allowing reasoning about scientific objects, it is critical to provide a validation process, therefore to use a decidable language to express statements. OWL constructs include mechanisms for identifying resources (RDF schema features), property restrictions (e.g. quantifications such as allValuesFrom), class interaction (e.g. intersection Of), comparison operations (e.g. equivalentClass), restricted cardinality (e.g. minCardinality), versioning (e.g. priorVersion), property characteristics (e.g. SymmetricProperty) and annotations. All terms cited are OWL constructs as specified in the OWL Guide (http://www.w3.org/TR/owl-guide). This set of constructs allows the reasoning about resources and thus can be exploited to achieve meaningful data integration.

Many technical issues underlying the Semantic Web, e.g. automated deduction and graph-based knowledge representations, have been studied extensively. The importance of the Semantic Web lies in the adoption of standards for metadata, knowledge representation and reasoning, and thus the potential leverage for joint tools and services. For more information on the Semantic Web, see http://www.w3.org/2001/sw and http://en.wikipedia.org/wiki/Semantic_Web.

6 Scientific Workflows

Scientific workflows are becoming recognized as an important unifying mechanism for combining scientific data management, analysis, simulation and visualization tasks, as witnessed by the following recent examples:

Meetings

- Scientific Data Management Framework Workshop, Argonne National Labs, August 2003, http://sdm.lbl.gov/∼arie/sdm/SDM.Framework.wshp.htm
- Sixth Biennial Ptolemy Miniconference, Featuring the Kepler Project, May 2005, UC Berkeley;
- LINK-Up Workshop on Scientific Workflows, San Diego Supercomputer Center, October 2004, http://kbis.sdsc.edu/events/link-up-11-04

Workshops

- Workflow in Grid Systems Workshop, GGF10, Berlin, March 2004, http://www.extreme.indiana.edu/groc/Worflow-call.htm
- e-Science Grid Environments Workshop, e-Science Institute, Edinburgh, May 2004, http://www.nesc.ac.uk/esi/events

Special issues devoted to the topic [27, 54].

Roughly speaking, the goal of scientific workflows is to capture scientific data handling, processing, and visualization steps in a way that facilitates repeated execution (often with different parameter settings or varying input files), possibly "steered" and guided by runtime user interactions. Other goals include reuse of workflows and workflow components (tasks, modules, steps, or actors), documentation and sharing of workflows, and capturing data provenance of the products created through a workflow. The various aspects of data integration, as discussed above, can often be considered as specific (often upstream) tasks of a larger workflow or data analysis pipeline, with downstream analysis and visualization components. In a sense, scientific workflows complement the more "static" *data integration* obtained by wrapping scientific data sources at the schema or conceptual level (e.g. via ontologies) and resulting in integrated or mediated data sources, by providing a dynamic *process integration* which includes elements of data integration at various stages or steps, but which also encompasses the overall scientific data management and analysis workflows that scientists need to consider as part of their scientific investigations.

6.1 Example: Promoter Identification Workflow (PIW)

Figure 8 shows a high-level, conceptual view of a typical scientific "knowledge discovery" workflow that links genomic biology techniques such as microarrays with bioinformatics tools such as BLAST to identify and characterize eukaryotic promoters – we call this the PIW (see also Refs. [53, 64,

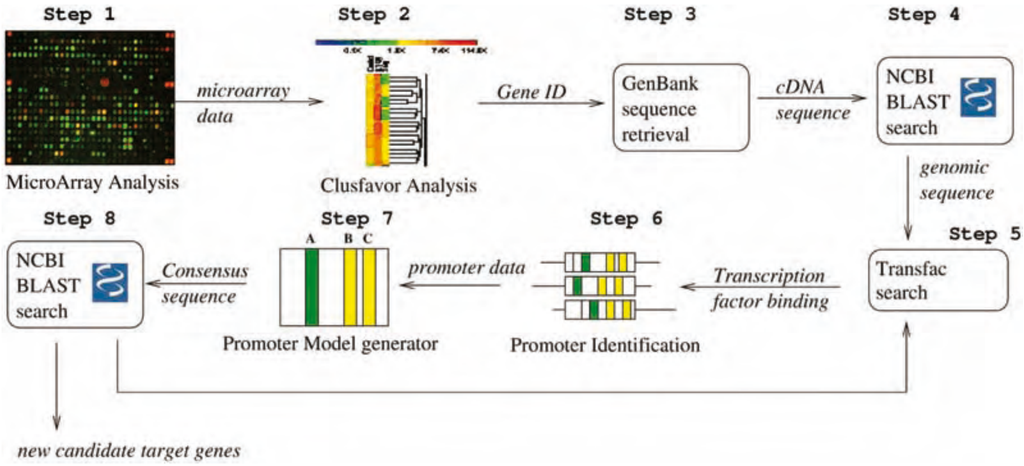


Figure 8 Conceptual ("napkin drawing") view of a PIW.

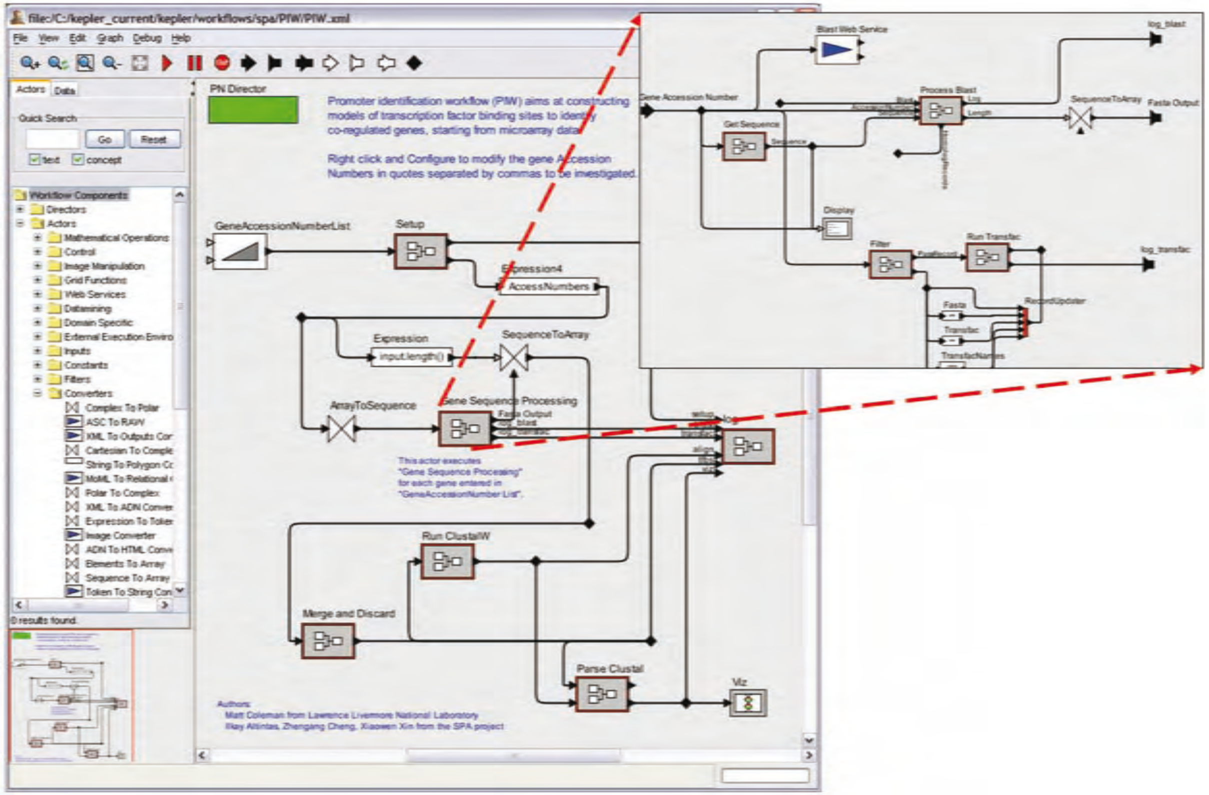
87]). Starting from microarray data, cluster analysis algorithms are used to identify genes that share similar patterns of gene expression profiles that are then predicted to be coregulated as part of an interactive biochemical pathway. Given the gene identifiers, gene sequences are retrieved from a remote database (e.g. GenBank) and fed to a tool (e.g. BLAST) that finds similar sequences. In subsequent steps, transcription factor-binding sites and promoters are identified using specialized tools to create a promoter model that can be iteratively refined. While Figure 8 leaves many details open, some features of scientific workflows can already be identified: there are a number of existing databases (such as GenBank) and computational tools (such as Clusfavor and BLAST) that need to be combined in certain ways to create the desired workflow. Even at this "napkin-drawing" level, scientific workflows are often data-centric, with edges corresponding to the dataflow between different computational components or databases. Unlike conventional (e.g. Unix) pipelines, scientific workflows may involve loops (such as the one from Step 8 to Step 5 in Figure 8) and often include explicit user interaction, e.g. to determine revised parameter settings, threshold values, etc., or the selection of data objects to be passed on to downstream steps.

In the past, accessing remote resources often meant implementing a wrapper that mimics a human entering the input of interest, submitting an HTML form, and "screen-scraping" the result from the returned page [52]. Today, more and more tools and databases become accessible via web services, avoiding low-level screen-scraping steps. Another trend is the development of web portals such as NCBI (http://www.ncbi.nlm.nih.gov). Nevertheless, chaining together workflow components, whether they are web services or other applications that are invoked, e.g. through shell commands or scripts, creates a number of problems that are similar to those found in data integration and thus can be similarly addressed, e.g. the various components are typically developed by independent parties and thus often do not use the same exchange format. Even if a common exchange (meta-)language such as XML is used, the different concrete XML schemata or DTDs being used require additional data transformation and schema mappings steps, such that the output of component $n - 1$ fits the structure of the input required for component n. Special intermediate components that can reconcile structural and/or semantic heterogeneities are sometimes called *shims* [41] or *adapters* [12] and can be seen as the workflow equivalents of wrappers or mediators in "conventional" data integration.

Figure 9 depicts a snapshot of an implementation of (part of) PIW in Kepler (http://kepler-project.org) [53]. Kepler is an extension of the Ptolemy II system specifically designed for scientific workflows. The Ptolemy project studies modeling, simulation and design of concurrent, real-time, embedded systems with a focus on assembly of concurrent components (http://ptolemy.eecs.berkeley.edu/ptolemyII). Using Ptolemy II terminology, we call the individual workflow steps *actors*, since they act as completely independent components which communicate with each other only through the dataflow channels indicated in Figure 9. A mechanism for collapsing details of a sub-workflow into an abstract component (called *composite actor* in Kepler) is essential to tame complexity. The upper-right window in Figure 9 has well-defined input and output ports, and thus corresponds to a (sub)-workflow that can be collapsed into a more abstract, composite actor (called "Gene-Sequence-Processing") as shown. Also, backward loops as the one in Figure 8 can often be avoided by incorporating higher-order collection programming constructs (known from functional programming, e.g. "map" or "fold") in the visual workflow language, resulting in a more comprehensible overall workflow model.

6.2 Scientific Workflow Requirements and Desiderata

Many scientific workflows exhibit a number of common "stages" or steps such as the following: (i) discovery of data sets, often by browsing and searching based on metadata of the data sets, (ii) querying and retrieval of the relevant data sets or parts of them, (iii) application of an "analysis pipeline" – in one or more analytical steps, e.g. cluster analysis – to the data sets, (iv) visualization and comparison of the results by the user, (v) repetition of some or all of the above, possibly with changed parameters and/or data sets until the user achieves a satisfactory result, and (vi) registration and storage of the analysis results along with workflow execution metadata (e.g. to facilitate repeatability of a workflow run).



Promoter identification workflow (PIW) aims at constructing models of transcription factor binding sites to identify co-regulated genes, starting from microarray data.

Right click and Configure to modify the gene Accession Numbers in quotes separated by commas to be investigated.

This actor executes "Gene Sequence Processing" for each gene entered in "GeneAccessionNumber List".

Authors:
Matt Coleman from Lawrence Livermore National Laboratory
Ilkay Altintas, Zhengang Cheng, Xiaowen Xin from the SPA project

Figure 9 Executable PIW in Kepler [53]. The composite actor (center) contains a nested sub-workflow (upper right). Workflow steps include remote service invocation and data transformation and analysis steps. An overall model of execution is enforced by a Process Network (PN) director (green box, upper left).

Scientific workflows often exhibit particular “traits”, e.g. they can be data intensive, computer intensive, analysis intensive and visualization intensive [53]. Depending on the intended user group, one might want to hide or emphasize particular aspects and technical capabilities of scientific workflows. For example, a computational biologist with extreme computational requirements and producing very large volumes of data might be interested in low-level workflow aspects such as data movement and remote job control. Thus, having workflow components (or actors) that operate at this level will be beneficial to the computational scientist. Conversely, a scientific workflow system should hide such aspects from analytical scientists who do not need specialized data and cycle management.

From a technical viewpoint, the following requirements and desiderata are often found in connection with scientific workflows:

R1 Seamless access to resources and services. This is a very common requirement, and web services provide a first, simple mechanism for remote service execution and remote database access via service calls. However, web services are a simple solution to a simple problem. Harder problems, e.g. web service orchestration and third-party transfer, are not solved by “vanilla” web services alone.

R2 Service composition and reuse, and workflow design. Since web services emerge as the basic building blocks for distributed Grid applications and scientific workflows, the problem of service composition, i.e. how to compose simple services to perform complex tasks, has become an attractive research topic [80]. Among the different approaches are those that view service composition as an artificial intelligence planning problem [9], a query planning problem [55, 56] or a general design and programming problem. A related issue is how to design components such that they are easily reusable and not geared to only the specific applications that may have driven their original development. By employing an actor-oriented approach at the design level [12], but also flexible means for data transformations and data integration at the “plumbing” level, the reusability of workflows and workflow components can be improved.

R3 Scalability. Some workflows involve large volumes of data and/or require high-end computational resources, e.g. running a large number of parallel jobs on a cluster computer. To support such data-intensive and computer-intensive workflows, suitable interfaces to Grid middleware components (sometimes called Compute-Grid and Data-Grid, respectively) are necessary. For example, the Kepler system includes actors to launch and monitor Globus jobs and to issue Storage Resource Broker (SRB; http://www.sdsc.edu/srb) commands for that purpose.

R4 Detached execution. Long-running workflows require an execution mode that allows the workflow control engine to run in the background on a remote server, without necessarily staying connected to a user's client application that has started and is controlling workflow execution.

R5 Reliability and fault-tolerance. Some computational environments are less reliable than others. For example, a workflow that incorporates a new web service can easily "break", as the latter can often fail, change its interface or just become unacceptably slow. To make a workflow more resilient in an inherently unreliable environment, contingency actions must be specifiable, e.g. fail-over strategies with alternate web services.

R6 User interaction. Many scientific workflows require user decisions and interactions at various steps. For example, an improved version of PIW allows the user to inspect intermediate results and select and re-rank them before feeding them to subsequent steps. An interesting challenge is the need for user interaction in a detached execution. Using a notification mechanism the user might be asked to reconnect to the running instance and make a decision before the paused (sub-)workflow can resume.

R7 "Smart" reruns. A special kind of user interaction is the change of a parameter of a workflow or actor. For example, in a visualization pipeline or a long running workflow, the user might decide to change some parameters after inspecting intermediate or even final results. A "smart" rerun does not execute the workflow from scratch, but only those parts that are affected by the parameter change. In dataflow-oriented systems (e.g. visualization pipeline systems such as AVS, OpenDX, SCIRun or Kepler) this is easier to realize than in more control-oriented systems (e.g. business workflow systems), since data and actor dependencies are already explicit in the system. Another useful technique in this context is *checkpointing*, which allows for backtracking to a previously saved state, e.g. in the case of a parameter change or even a system failure, without starting over from scratch.

R8 "Smart" (semantic) links. A scientific workflow system should assist workflow design and data binding phases by suggesting which actor components might possibly fit together (this is also an aspect of *R*2) or by indicating which data sets might be fed to which actors or workflows. To do so, some of the semantics of data and actors has to be captured. However, capturing data semantics is a hard problem in many scientific disciplines, e.g. measurement contexts, experimental protocols and assumptions made are often not adequately represented. Even if corresponding metadata are available, it is often not clear how to best make it useable by the system. It seems clear though that ontologies provide a very useful semantic type system for scientific workflows, in addition to the current (structural) type systems [5, 11, 12].

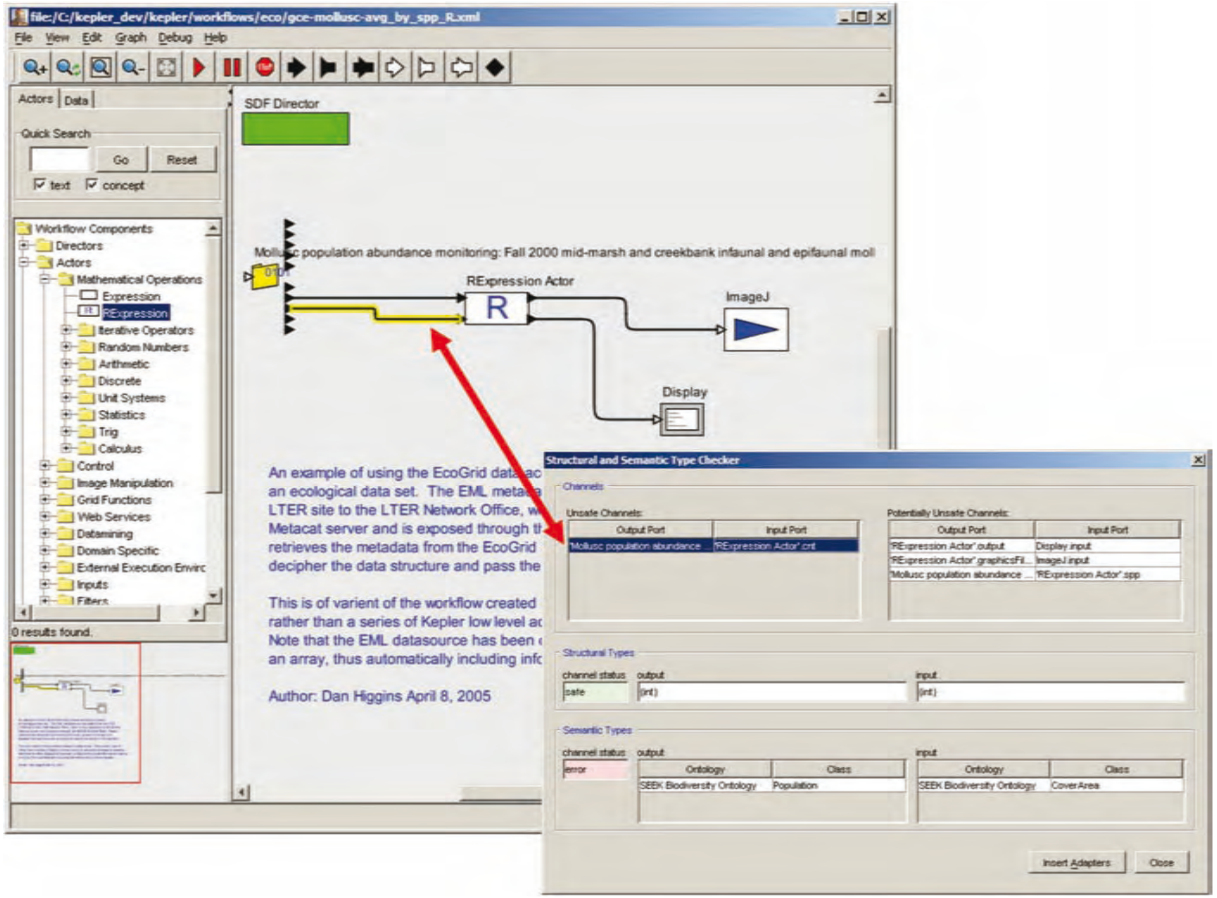
R9 Data provenance. Just as the results of a conventional wet lab experiment should be reproducible, computational experiments and runs of scientific workflows should be reproducible, and indicate which specific data products and tools have been used to create a derived data product. Beyond the conventional capture of metadata, a scientific workflow system should be able to automatically log the sequence of applied steps, parameter settings and (persistent identifiers of) intermediate data products. A related desiderata is automatic report generation: The system should allow the user to generate reports with all relevant provenance and runtime information, e.g. in XML format for archival and exchange purposes, and in HTML (generated from the former, e.g. via an XSLT script) for human consumption.

6.3 Semantic Extensions and Scientific Workflow Design

One problem in designing scientific workflows lies in the increasing amount of components (and possibly sub-workflows) made available, e.g. as web services, SOAP-lab services [74], statistics packages (e.g. R packages) or as other custom applications. While web services provide a minimalistic interface definition of the various operations via the accompanying WSDL files, there are no standard means of helping the scientist or a scientific workflow system to determine whether two components/web service operations can be chained together. One promising approach is to separate the concerns of structural data typing and semantic data typing [5,12], i.e. to use conventional structural typing mechanisms (such as XML DTDs or schemata) to describe structural aspects of the data flowing between workflow components, but to use a *semantic typing* mechanism, such as an OWL concept expression, to describe what kind of data (at the conceptual level) is being exchanged.

Consider, for example, the Kepler workflow shown in Figure 10. A connection has been identified to be structurally safe – an array of integers is produced and consumed by the two connected actors – but semantically unsafe – the type checking window of the highlighted connection indicates this with a green, respectively red type status, next to the display of the structural and semantic port types. Note that to guarantee semantic type safety, (i) semantic types (concept expressions) have to be declared for actor ports and (ii) a reasoning engine has to establish that the desired concept subsumption relation holds. For structural type safety any of the common type systems can be used, e.g. XML DTD, XML Schema, or even object-oriented or relational schemata. For semantic types, concept expressions, e.g. from a DL ontology, encoded in OWL can be used. The simple workflow in Figure 10, for example, uses concepts from a biodiversity ontology Science Environment for Ecological Knowledge (SEEK) project web site (http://seek.ecoinformatics.org).

Many other workflow systems have been developed by academic institutions and industry to address (often partially) the challenges presented above. Among those designed to support scientific protocols, academic systems include Triana an open-source problem-solving environment de-



An example of using the EcoGrid data ac
an ecological data set. The EML meta
LTER site to the LTER Network Office, w
Metacat server and is exposed through th
retrieves the metadata from the EcoGrid
decipher the data structure and pass the

This is of varient of the workflow created
rather than a series of Kepler low level ac
Note that the EML datasource has been
an array, thus automatically including info

Author: Dan Higgins April 8, 2005

Channels

Unsafe Channels:

Output Port	Input Port
'Mollusc population abundance ...	'RExpression Actor'.cnt

Potentially Unsafe Channels:

Output Port	Input Port
'RExpression Actor'.output	Display.input
'RExpression Actor'.graphicsFil...	ImageJ.input
'Mollusc population abundance ...	'RExpression Actor'.spp

Structural Types

channel status	output	input
safe	{int}	{int}

Semantic Types

channel status: error

output:

Ontology	Class
SEEK Biodiversity Ontology	Population

input:

Ontology	Class
SEEK Biodiversity Ontology	CoverArea

Figure 10 A Kepler data set actor (yellow "folder") is connected with an R statistics actor. The Kepler type checker determines that the connection is safe with respect to the *structural* type (array of integers), but is *semantically* not well-typed since the output semantic type "Population" is not compatible with (subsumed by) the input semantic type "Cover Area" of the instantiated R actor.

veloped at Cardiff University that combines an intuitive visual interface with powerful data analysis tools (http://www.trianacode.org/index.html), Kepler (based on the Ptolemy II system for heterogeneous, concurrent modeling and design, http://kepler-project.org) criteria and Taverna (developed in the UK by a European team; http://taverna.sourceforge.net and http://homepages.cs.ncl.ac.uk/peter.li/home.formal/tutorial).

7 Conclusion

Data integration is a critical issue for life science. Although many traditional approaches such as data warehouses, federations and mediations provide efficient integrated platform to query the data, there is a need for a broader view on biological data integration. Although scientists need to integrate data from multiple heterogeneous and autonomous sources, they also need to integrate the various applications made available to them. Traditional database integration approaches do not offer solutions to the complex problem of application integration. Today, the community seems to agree on one fundamental fact: to enable efficient and meaningful integration, biological resources – databases and applications alike – need to provide semantic information. Many solutions are being developed to define, support and exploit this semantic layer for data integration. Orthogonally, traditional database approaches do not provide an access to data suitable to scientific protocols. To allow scientists to express and execute scientific queries, systems based on workflows are developed. This work in progress should result in a new generation of data integration systems.

References

1 ABITEBOUL, S., P. BUNEMAN AND D. SUCIU. 1999. *Data on the Web: From Relations to Semistructured Data and XML*. Morgan Kaufmann, San Francisco, CA.

2 BACHMAN, C. 1969. Data structure diagrams. DataBase: A Quarterly Newsletter of SIGBDP **1(2)**: 4–10.

3 BACHMAN, C. 1972. The evoluation of storage structures. Commun. ACM **15(15)**: 628–34.

4 BECHHOFER, S., I. HORROCKS, C. GOBLE AND R. STEVENS. 2001. OilEd: a reasonable ontology editor for the semantic web. Lecture Notes Artif. Intell.: 396–408.

5 BERKLEY, C., S. BOWERS, M. JONES, B. LUDÄSCHER, M. SCHILDHAUER AND J. TAO. 2005. Incorporating semantics in scientific workflow authoring. In Proc. Int. Conf. on Scientific and Statistical Database Management: 75–8.

6 BERNERS-LEE, T. 2000. *Weaving the Web*. Harper San, Francisco, CA.

7 BERNSTEIN, P. A. 2003. Applying model management to classical meta data problems. In Proc. Biennial Conf. on Innovative Data Systems Research.

8 BISKUP, J. 1998. Achievements of relational database schema design theory revisited. Lecture Notes Comput. Sci. **1358**: 29–54.

9 Blythe, J., E. Deelman and Y. Gil. 2003. Planning for workflow construction and maintenance on the grid. In Proc. Int. Conf. on Automated Planning and Scheduling.

10 Booch, G., J. Rumbaugh and I. Jacobson. 1998. *The Unified Modeling Language User Guide*. Addison Wesley, Reading, MA.

11 Bowers, S., K. Lin and B. Ludäscher. 2004. On integrating scientific resources through semantic registration. In Proc. 16th Int. Conf. on Scientific and Statistical Database Management: 349–52.

12 Bowers, S. and B. Ludäscher. 2005. Actor-oriented design of scientific workflows. Lecture Notes Comput. Sci. **3716**: 369–84.

13 Broekstra, J., A. Kampman and F. v. Harmelen. 2002. Sesame: a generic architecture for storing and querying rdf and rdf schema. Lecture Notes Comput. Sci. **2342**: 54–68.

14 Bronts, G. H. W. M., S. J. Brouwer, C. L. J. Martens and H. A. Proper. 1995. A unifying object role modelling approach. Inf. Syst. **20**: 213–35.

15 Burks, C. 1999. Molecular biology database list. Nucleic Acids Res. **27**: 1–9.

16 Cattell, R. G. G., D. Barry, M. Berler, et al. 2000. *The Object Data Standard: ODMG 3.0*. Morgan Kaufmann, San Francisco, CA.

17 Chaudhri, B., A. Raschid and R. Zicari. 2003. *XML Data Management: Native XML and XML-Enabled Database Systems*. Addison Wesley Professional, Reading, MA.

18 Chaudhuri, S. and U. Dayal. 1997. An overview of data warehousing and OLAP technology. ACM SIGMOD Record **26**: 65–74.

19 Chen, I.-M. A. and V. M. Markowitz. 1995. An overview of the Object-Protocol Model (OPM) and OPM data management tools. Inf. Syst. **20**: 393–418.

20 Chen, J., S. Chung and L. Wong. 2003. The Kleisli query system as a backbone for bioinformatics data integration and analysis. In Lacroix, Z. and T. Critchlow (eds.), *Bioinformatics: Managing Scientific Data*. Morgan Kaufmann, San Francisco, CA: 147–88.

21 Chen, P. 1976. The entity–relationship model – toward a unified view of data. ACM Trans. Database Syst. **1**: 9–36.

22 Cheung, K. H., K. Y. Yip, A. Smith, R. Deknikker, A. Masiar and M. Gerstein. 2005. YeastHub: a semantic web use case for integrating data in the life sciences domain. Bioinformatics **21**: i85–96.

23 Ciaramita, M., A. Gangemi, E. Ratsch, J. Saric and I. Rojas. 2005. Unsupervised learning of semantic relations between concepts of a molecular biology ontology. In Proc. 19th Int. Joint Conf. on Artificial Intelligence. http://ijcai.org/search.php.

24 Cohen-Boulakia, S., S. Davidson and C. Froidevaux. 2005. A user-centric framework for accessing sources and tools. Lecture Notes Bioinform. **3615**: 3–18.

25 Curbera, F., M. Duftler, R. Khalaf, W. Nagy, N. Mukhi and S. Weerawarana. 2002. Unraveling the Web Services web: an introduction to SOAP, WSDL, and UDDI. Internet Comput. **6**: 86–93.

26 Davidson, S. B., C. Overton, V. Tannen and L. Wong. 1997. BioKleisli: a digital library for biomedical researchers. Int. J. Digital Libraries **1**: 36–53.

27 Deelman, E. and I. Taylor. 2005. Special Issue on Scientific and Grid Workflows. J. Grid Comput. **3**: 151.

28 Doan, A., P. Domingos and A. Y. Halevy. 2003. Learning to match the schemata of data sources: a multistrategy approach. Machine Learn. **50**: 279–301.

29 Engels, G., M. Gogolla, U. Hohenstein, K. Hülsmann, P. Löhr-Richter, G. Saake and H.-D. Ehrich. 1992. Conceptual modelling of database applications using an extended ER model. Data Knowledge Eng. **9**: 157–204.

30 Etzold, T. and P. Argos. 1993. SRS: an indexing and retrieval tool for flat file

data libraries. Comput. Applic. Biosci. **9**: 49–57.

31 Etzold, T., H. Harris and S. Beaulah. 2003. SRS: an integration platform for databanks and analysis tools in bioinformatics. In Lacroix, Z. and T. Critchlow (eds.), *Bioinformatics: Managing Scientific Data*. Morgan Kaufmann, San Francisco, CA: 109–46.

32 Fagin, R., P. Kolaitis, R. J. Miller and L. Popa. 2003. Data exchange: semantics and query answering. In Proc. 9th Int. Conf. on Database Theory: 207–24.

33 Galperin, M. Y. 2005. The molecular biology database collection: 2005 update. Nucleic Acids Res. **33**: D5–24.

34 Gambin, A., J. Hidders, N. Kwasnikowska, S. Lasota, J. Sroka, J. Tyszkiewicz and J. Van den Bussche. 2005. NRC as a formal model for expressing bioinformatics workflows. Proc. ISMB **13**: poster presentation.

35 Gangemi, A., D. M. Pisanelli and G. Steve. 1999. An overview of the ONIONS project: applying ontologies to the integration of medical terminologies. Data Knowledge Eng. **31**: 183–220.

35a Goble, C.A. and R. D. Stevens. 2001. Managing Biological Information Using Biological Knowledge. In NETTAB – Network Tools and Applications in Biology CORBA and XML: Towards a Bioinformatics Integrated Network Environment. Advanced Biotechnology Center, Genoa, Italy.

36 Goble, C., S. Pettifer and R. Stevens. 2003. myGrid: *in silico* experiments in bioinformatics. In Foster, I. and C. Kesselman (eds.), *The Grid: Blueprint for a New Computing Infrastructure*, Morgan Kaufmann, San Francisco, CA.

37 Gruber, T. R. 1993. A translation approach to portable ontologies. Knowledge Acquisition **5**: 199–220.

38 Haas, L., B. Eckman, P. Kodali, J. R. E. Lin and P. Schwarz. 2003. DiscoveryLink. In Lacroix, Z. and T. Critchlow (eds.), *Bioinformatics: Managing Scientific Data*. Morgan Kaufmann, San Francisco, CA: 303–34.

39 Haas, L., M. A. Hernández, H. Ho, L. Popa and M. Roth. 2005. Clio grows up: from research prototype to industrial tool. In Proc. Int. Conf. on Management of Data: 805–10.

40 Hu, J., C. Mungall, D. Nicholson and A. Archibald. 1998. Design and implementation of a CORBA-based genome mapping system prototype. Bioinformatics **14**: 112–20.

41 Hull, D., R. Stevens, P. Lord, C. Wroe and C. Goble. 2004. Treating shimantic web syndrome with ontologies. In Proc. First AKT Workshop on Semantic Web Services.

42 Hull, R. and R. King. 1987. Semantic database modeling: survey, applications, and research issues. ACM Comput. Surv. **19**: 201–60.

43 Kennedy, J. B., R. Kukla and T. Paterson. 2005. Scientific names are ambiguous as identifiers for biological taxa: their context and definition are required for accurate data integration. Lecture Notes Comput. Sci. **3615**: 80–95.

44 Kikuchi, N., A. Kameyama, S. Nakaya, H. Ito, T. Sato, T. Shikanai, Y. Takahashi and H. Narimatsu. 2005. The carbohydrate sequence markup language (CabosML): an XML description of carbohydrate structures. Bioinformatics **21**: 1717–8.

45 Kosky, A., I.-M. A. Chen, V. Markowitz and E. Szeto. 1998. Exploring heterogeneous biological databases: tools and applications. Lecture Notes Comput. Sci. **1377**: 499–513.

46 Krummenacker, M., S. Paley, L. Mueller, T. Yan and P. D. Karp. 2005. Querying and computing with BioCyc databases. Bioinformatics **21**: 3454–55.

47 Kulak, D. and E. Guiney. 2000. *Use Cases – Requirements in Context*. Addison Wesley, Reading, MA.

48 Lacroix, Z. 2003. Web data retrieval and extraction. Data Knowledge Eng. **44**: 347–367.

49 Lacroix, Z. and T. Critchlow (eds.). 2003. *Bioinformatics: Management of Scientific Data*. Morgan Kaufmann, San Francisco, CA.

50 LACROIX, Z., K. PAREKH, M.-E. VIDAL, M. CARDENAS AND N. MARQUEZ. 2005. BioNavigation: selecting optimum paths through biological resources to evaluate ontological navigational queries. Lecture Notes Comput. Sci. **3615**.

51 LARMAN, C. 1998. *Applying UML and Patterns*. Prentice-Hall, Upper Saddle River, NJ.

52 LIU, L., C. PU AND W. HAN. 2001. An XML enabled data extraction tool for web sources. Int. J. Information Syst. (Special Issue on Data Extraction, Cleaning, and Reconciliation) **26**: 563–83.

53 LUDÄSCHER, B., I. ALTINTAS, C. BERKLEY, et al. 2006. Scientific workflow management and the Kepler system. Concurrency and Computation: Practice and Experience.

54 LUDÄSCHER, B. AND C. GOBLE (eds.). 2005. ACM SIGMOD-Record (Special Section on Scientific Workflows) **34**.

55 LUDÄSCHER, B., A. GUPTA AND M. E. MARTONE. 2003. A model-based mediator system for scientific data management. In LACROIX, Z. AND T. CRITCHLOW (eds.), *Bioinformatics: Managing Scientific Data*. Morgan Kaufmann, San Francisco, CA.

56 LUDÄSCHER, B. AND A. NASH. 2004. Web service composition through declarative queries: the case of conjunctive queries with union and negation. In Proc. 20th Int. Conf. on Data Engineering, http://www.informatik.uni-trier.de/~ley/db/conf/icde/icde2004.html.

57 LUDÄSCHER, B. AND L. RASCHID (eds.). 2005. Proceedings of the 2nd International Workshop on Data Integration in the Life Sciences (DILS). Lecture Notes Comput. Sci. **3615**.

58 MARKOWITZ, V. M., J. CAMPBELL, I.-M. A. CHEN, A. KOSKY, K. PALANIAPPAN AND T. TOPALOGLOU. 2003. Integration challenges in gene expression data management. In LACROIX, Z. AND T. CRITCHLOW (eds.), *Bioinformatics: Managing Scientific Data*. Morgan Kaufmann, San Francisco, CA.

59 MCGUINNESS, D. L. 2000. Conceptual modeling for distributed ontology environments. Lecture Notes Comput. Sci. **1867**.

60 MUSEN, M. A., S. W. TU, H. ERIKSSON, J. H. GENNARI AND A. R. PUERTA. 1993. PROTEGE-II: an environment for reusable problem-solving methods and domain ontologies. In Proc. Int. Joint Conf. on Artificial Intelligence.

61 NAMBIAR, U., Z. LACROIX, S. BRESSAN, M.-L. LEE AND Y. G. LI. 2002. Current approaches to XML management. IEEE Internet Comput. **6**: 43–51.

62 NOY, N. AND M. MUSEN. 2000. PROMPT: algorithm and tool for automated ontology merging and alignment. In Proc. Natl Conf. on Artificial Intelligence: 450–55.

63 OINN, T., M. ADDIS, J. FERRIS, et al. 2004. Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics **20**: 3045–54.

64 PETERSON, L., E. YIN, D. NELSON, I. ALTINTAS, B. LUDÄSCHER, T. CRITCHLOW, A. J. WYROBEK AND M. A. COLEMAN. 2003. Mining the frequency distribution of transcription factor binding sites of ionizing radiation responsive genes. In Proc. New Horizons in Genomics Conf.

65 PISANELLI, D. M., A. GANGEMI AND G. STEVE. 1999. A medical ontology library that integrates the UMLS metathesaurus. In Proc. Joint Eur. Conf. on Artificial Intelligence in Medicine and Medical Decision Making: 239–48.

66 REBHAN, M., V. CHALIFA-CASPI, J. PRILUSKY AND D. LANCET. 1997. GeneCards: encyclopedia for genes, proteins and diseases. Trends Genet. **13**: 163.

67 REDASCHI, N., K. KRUSZEWSKA, P. LIJNZAAD AND P. RODRIGUEZ-TOME. 1998. Accessing the EMBL database through CORBA – implementation of a Browsing Server (EMCORBA v2). Proceedings of the German Conference on Bioinformatics (GCB). Bioinformatics **14**: 656–64.

68 ROBBINS, R. J. 2004. Object identity and life science research [Position paper]. In Proc. W3C Workshop on Semantic Web in the Life Sciences.

69 Rodriguez-Tome, P., C. Helgesen, P. Lijnzaad and K. Jungfer. 1997. A CORBA server for the Radiation Hybrid DataBase. Proc. ISMB **5**: 250–3.

70 Roth, M. A., H.F. Korth and A. Silberschatz. 1988. Extended algebra and calculus for nested relational databases. ACM Trans. Database Syst. **13**: 389–417.

71 Roux-Rouquie, C. M., N. Caritey, L. Gaubert and C. Rosenthal-Sabroux. 2004. Using the Unified Modelling Language (UML) to guide the systemic description of biological processes and systems. Biosystems **75**: 3–14.

72 Rumbaugh, J., I. Jacobson and G. Booch. 1999. *The Unified Modeling Language Reference Guide*. Addison Wesley, Reading, MA.

73 Salamone, S. 2004. LSID: an informatics lifesaver. Bio-IT World: http://www.bio-itworld.com/archive/011204/lsid.html.

74 Senger, M., P. Rice and T. Oinn. 2003. SoapLab –a unified Sesame door to analysis tools. In Proc. UK e-Science All Hands Meeting.

75 Stein, L. 2002. Creating a bioinformatics nation. Nature **417**: 119–20.

76 Stevens, R., C. Goble, N. Paton, S. Bechhofer, G. Ng, P. Baker and A. Brass. 2003. Complex query formulation over diverse information sources in TAMBIS. In Lacroix, Z. and T. Critchlow (eds.), *Bioinformatics: Managing Scientific Data*. Morgan Kaufmann, San Francisco, CA: 189–224.

77 Stevens, R., C. Goble, P. Baker and A. Brass. 2001. A classification of tasks in bioinformatics. Bioinformatics **17**: 180–8.

78 Stevens, R. and C. Miller. 2000. Wrapping and interoperating bioinformatics resources using CORBA. Brief. Bioinform. **1**: 9–21.

79 Stevens, R., C. Wroe, P. Lord and C. Goble. 2003. Ontologies in bioinformatics. In Staab, S. and R. Studer (eds.), *Handbook on Ontologies in Information Systems*. Springer, Berlin: 635–57.

80 Tannen, V., S. Davidson and S. Harker. 2003. The Information Integration System K2. In Lacroix, Z. and T. Critchlow (eds.), *Bioinformatics: Managing Scientific Data*. Morgan Kaufmann, San Francisco, CA: 225–48.

81 The Gene Ontology Consortium. 2000. Gene Ontology: tool for the unification of biology. Nat. Genet. **25**: 25–9.

82 Torlone, R. 2003. *Conceptual Multidimensional Models, Multidimensional Databases: Problems and Solutions*. Idea Group: 69–90.

83 Torlone, R. and L. Cabibbo. 2004. on the integration of autonomous data marts. In Proc 6th Int. Conf. on Scientific and Statistical Database Management.

84 Velegrakis, Y., R. Miller and J. Mylopoulos. 2005. Representing and querying data transformations. In Proc. Int. Conf. on Data Engineering: 81–92.

85 Vinoski, S. 1997. CORBA: Integrating diverse applications within distributed heterogeneous environments. Commun. Mag. **14**: 46–55.

86 Wain, H. M., M. Lush, F. Ducluzeau and S. Povey. 2002. Genew: the human gene nomenclature database. Nucleic Acids Res. **30**: 169–71.

87 Werner, T. 2001. Target gene identification from expression array data by promoter analysis. Biomol. Eng. **17**: 87–94.

88 Westbrook, J., N. Ito, H. Nakamura, K. Henrick and H. M. Berman. 2005. PDBML: the representation of archival macromolecular structure data in XML. Bioinformatics **21**: 988–92.

89 Wiederhold, G. 1992. Mediators in the architecture of future information systems. IEEE Comput. **25**: 38–49.

90 Wiederhold, G. and M. Genesereth. 1997. The conceptual basis for mediation services. IEEE Expert Intell. Syst. Applic. **12**: 38–47.

91 Wilkinson, K., C. Sayers, H. A. Kuno and D. Reynolds. 2003. Efficient RDF storage and retrieval in Jena2. In Proc. First Int. Workshop on Semantic Web and Databases (co-located with VLDB): 131–50.

92 Wolstencroft, K., A. Brass, I. Horrocks, P. Lord, U. Sattler, D. Turi and R. Stevens. 2005. A little semantic web goes a long way in biology. Lecture Notes Comput. Sci. **3729**: 786–800.

43
Visualization of Biological Data

Harry Hochheiser, Kevin W. Eliceiri, and Ilya G. Goldberg

1 Introduction

Despite their considerable power, the computational techniques described throughout this book are unlikely to replace direct visual examination of biological data. High-resolution, interactive displays leverage the considerable processing power of the human visual system to help biologists identify patterns, locate outliers and build hypotheses that would escape detection by less-sensitive algorithmic approaches.

This chapter will examine two classes of visualizations of biological data, and discuss some of the issues related to their use and interpretation. The first class of data is generated by direct light-based microscopy imaging techniques, which provide pictures and movies of biological phenomena. Although traditional two-dimensional (2-D) images may be relatively straightforward to examine visually, 5-D (x,y,z, time and wavelength) data present substantial visualization challenges. These advanced multidimensional microscopy techniques have proven to be powerful biological research techniques, allowing scientists to record such phenomena as the dynamic process of embryogenesis. Increasingly, these methods are beginning to show great promise for medical imaging. While histopathology-based techniques are still the mainstay microscopy techniques in medicine, light microscopy methods such as fluorescence and multiphoton (MP) microscopy are being explored to track key medical processes such as cancer metastasis. There is promise that some light-microscopy modalities might be developed into diagnostic methods to complement the more traditional medical imaging techniques such as positron emission tomography (PET) and magnetic resonance imaging (MRI).

The second class of visualization discussed in this chapter includes data that are not amenable to the direct visual display used for microscopy data. Despite the possibility of graphical depictions of DNA helices, genomic sequences are most easily interpreted as strings in the familiar four-letter alphabet. Derived data such as sequence alignments, high-dimensional repre-

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

sentations of microarray gene expression data, pathways and other computationally derived data sets can also benefit from displays designed to highlight trends and patterns. Drawing from recent work in the young field of *information visualization*, systems for analyses of these data types use techniques like interactive filtering and multiple coordinated displays to assist users. This chapter surveys tools that apply these techniques to bioinformatics data, including descriptions of systems for examination of genomic sequences, microarray data, proteomics, interaction networks and pathways, phylogenies and taxonomies, and phenotypes and lineages.

Recent advances in the collection and storage of large collections of microscopy images have led to the advent of high-content screening (HCS), in which collections of cells can be systematically examined for responses to genetic and chemical stimuli. Management and interpretation of the resulting large volumes of data requires visualization of image data, metadata describing both images and experimentally meaningful collections of those images, and results of automated analyses. An examination of the image informatics requirements of HCS illustrates some of the challenges involved in building complete solutions.

Visualization of molecular structures is not discussed in this chapter. For recent work in this area, see Ref. [127].

2 Microscopy Image Visualization

Current cell-based screening methods have evolved from simple nonimaged light scattering or fluorescence measurements to cell imaging systems where cells in individual wells of a multi-well plate are imaged using wide-field fluorescence microscopy techniques. The challenge with these cell imaging systems is to develop robust algorithms that can automatically extract meaningful morphometric or spectral parameters from populations of cells. Due to the vastly increased amount of data available from imaging, these screening techniques are collectively called HCS. Several live cell imaging methods are discussed below. While all are not standard HCS techniques – often because of technical challenges – they are all proven methods to extract additional information on dynamic *in vivo* processes.

2.1 Fluorescence Microscopy Techniques Applicable to HCS Screening

The availability of extensive sequence data from many organisms, including humans, affords probes, such as fluorescent protein chimeras, to be made to identify any protein; these chimeras can identify the location and distribution of that protein within an organism. Furthermore, advanced optical techniques can yield information on the physiological state of a cell or the molecular environment of a particular reporter probe. The challenge for the HCS researcher is to harness all of these techniques in a manner that allows for the effective identification of specific changes in cell architecture or physiology. These techniques include the following.

2.1.1 Spectral Imaging

Fluorescence signals may be derived from probes such as fluorescent protein reporters or endogenous fluorophores such as collagen or reduced nicotinamide adenine dinucleotide [143] (Figure 1). Fluorescence spectroscopy studies using specimens in cuvettes have yielded a wealth of information about how subtle fluorescence spectral shifts may be used to report on molecular environments of fluorophores [110]. The availability of spectral analysis on a pixel-by-pixel basis in an image could extend these analyses to molecules in identified cellular or subcellular environments. Current cell-based HCS screening systems have very limited spectral discrimination, having only three or four spectral channels. Although this is adequate for identifying fluorophores with well-separated emission spectra, difficulties arise when spectra overlap or when it is necessary to measure the spectral shifts of certain indicator probes. Double or triple fluorescence-labeling approaches have great power to investigate complicated processes, as individual gene products of interest can be tagged with different colors. This approach is often complicated by overlapping emission spectra of many of the most popular fluorophores such as cyan, yellow and green fluorescent protein (CFP, YFP and GFP). Spectral imaging techniques can simplify multiple fluorescence label experiments by allowing for fast, flexible probe separation unavailable with traditional techniques of optical filter separation.

2.1.2 Lifetime Imaging

A fluorescence signal contains more information than just intensity and color. The lifetime of the excited state, which gives rise to the fluorescence signal, is diagnostic of the fluorophore and also of its microenvironment [11] (Figure 2). Factors such as ionic strength, hydrophobicity, oxygen concentration, binding to macromolecules and the proximity of molecules that can deplete the excited state by resonance energy transfer can all modify the lifetime of a fluorophore. Measurements of lifetimes can therefore be used as indicators of these parameters. Fluorescence lifetime measurements are generally absolute, being independent of the concentration of the fluorophore. Furthermore, lifetime properties may be particularly useful in identifying fluorophores with significantly overlapping spectral properties. Fluorescence lifetime imaging microscopy (FLIM) has been recently described and the potential of this

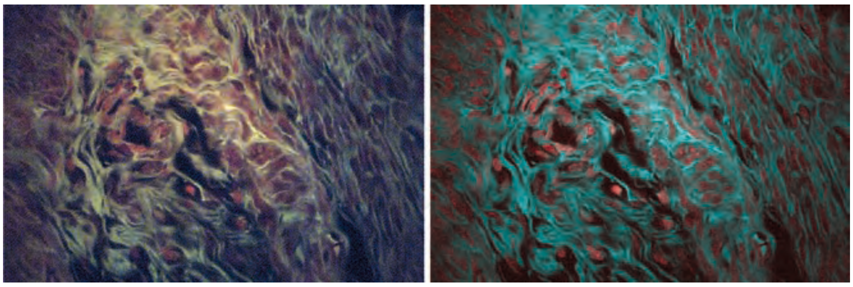


Figure 1 Spectral imaging. Seventeen-channel spectral image of a methyl green-stained section of uterus imaged with MP excitation. Displayed is image chrominance as a weighted function of wavelength values in each channel. The left-hand image is colored according to "best-guess" mapping, with the first third of the channels equally weighted toward red, the second third toward green, and the last third toward blue. The right-hand image illustrates the discrimination of nuclei by overweighting the contribution of channel 1 (660 nm) and negatively weighting channel 5 (600 nm) with respect to the red color component. This scheme reveals nuclei as red and surrounding tissue as turquoise. (Uterus section prepared by Al Kutchera, Midwest Microtech; reproduced from Eliceiri *et al.* 2005. Photochem. Photobiol. **81**: 1116 with permission of the American Society for Photobiology.)

technique demonstrated [66, 112, 190]. Practical applications include *in vivo* mapping of the metabolic states of a cell by using FLIM to distinguish the two forms (bound and free) of the intrinsic fluorophore NADH [16]. FLIM is not currently available in a commercial HCS instrument due to technical challenges such as the long photon counting times needed by FLIM. As the electronics and detectors for FLIM continue to improve, the speed of this technique should as well, making it more practical for HCS studies.

2.1.3 Fluorescence Resonant Energy Transfer (FRET)

FRET is a technique that can detect the close proximity of two fluorescently labeled molecules [116, 172]. Furthermore, this technique may be readily used *in vivo*. If one of the fluorophores (the donor) has a fluorescence emission spectrum that overlaps the excitation spectrum of the second fluorophore (the acceptor), then fluorescence from the donor can be partially quenched if the acceptor molecule is closer than the so-called Förster distance (around 10 nm). Energy is transferred to the acceptor molecule, which increases in fluorescence; the effect varying as the sixth power of the distance separating the two molecules. The technique is capable of detecting when a ligand binds to a receptor and so can be a potent way of visualizing the activation of a signal transduction pathway. However, often it is difficult to obtain reliable FRET estimations from measurements of fluorescence intensity changes, particularly within intracellular domains. Selective bleaching or compartmentalization of the two fluorophores can give rise to changes in relative fluorescence that are

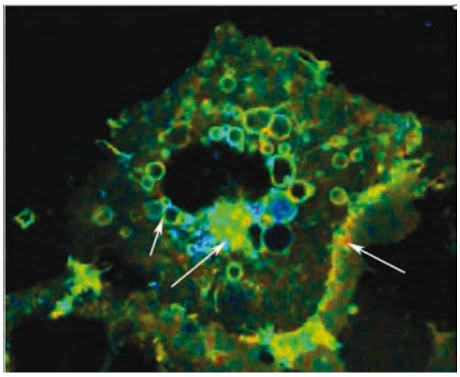


Figure 2 Lifetime imaging. GFP-R-Ras(38V), a constitutively active mutant of R-Ras, imaged at 900 nm using a Ti:sapphire laser and lifetime detector. Color mapping indicates changes in fluorescence lifetime Note more blue/green (shorter) lifetime values on vesicles (arrows near nucleus) and orange/green (longer) lifetime values at a membrane ruffle. Lifetime imaging can be used to reveal information about the dynamic interaction of labeled proteins or endogenous fluorescence in the context of its *in vivo* microenvironment. (Image courtesy of Dr Patricia Keely, University of Wisconsin–Madison.)

indistinguishable from those resulting from a FRET interaction. However, lifetime imaging can potentially circumvent these problems. When a FRET interaction occurs, the donor excited state lifetime increases. This is an absolute change that is independent of intensity. Using lifetime measurements to estimate FRET can potentially yield the percentage of the donor species that is involved in the FRET interaction (ratio of the shorter to the longer lifetime components) and the proximity if the two interacting molecules (degree of lifetime shortening). Practical applications of FRET include the investigation of protein–membrane interactions [118,119,197] and Ca^{2+} dynamics [187].

2.1.4 **Optical Sectioning**

Optical sectioning fluorescence microscopy has become the method of choice for imaging living specimens as it offers high signal-to-background ratios and the ability to spectrally discriminate between multiple fluorophores. Recently developed techniques such as confocal [137,193] or MP [51,194] imaging allow optical sections to be made of intact live specimens. These may be collected as stacks of images at different focal depths to obtain 3-D structural data. Stacks of images may be collected at regular time intervals in order to reveal the dynamics of 3-D structures in living tissue [182]. Optical sectioning could allow for cells in an HCS screen to be grown in a collagen gel, possibly a more natural environment for some cells. Three-dimensional imaging could be used to measure morphometric parameters of a cell. The association of cells with an extracellular matrix could be also characterized. HCS systems that use optical sectioning such as differential interference (DIC) are being widely used [94, 195]. These systems allow scientists to do high-content tracking of phenomena in 4-D. There is also active development to utilize laser-scanning methods such as confocal and MP laser-scanning microscopy that would allow for better viability and deeper sectioning [78, 111].

2.1.5 MP Imaging

MP laser-scanning microscopy (MPLSM) uses laser raster scanning to assemble an image (Figure 3). At a very high photon density, two or more photons may be simultaneously absorbed by an excitable molecule in order to generate fluorescence emission. The sum of the individual photon energies is equivalent to the energy transition of a single photon absorption event [51]. In the case of two-photon imaging, the excitation wavelength is set to about twice that of the absorption peak of the fluorophore being observed. Normally, this wavelength would not produce any appreciable fluorophore excitation. However, if a high-power, ultra-short pulse laser is used, it is possible to achieve instantaneous photon densities that will give rise to a significant yield of two-photon events at the focal volume of an objective lens, while maintaining a mean power level that will not damage the specimen. In this manner, fluorophore excitation is confined to the focal volume because the photon density is insufficient to generate appreciable MP events outside of this region. Optical sectioning is achieved because there is no appreciable fluorophore excitation above or below the focal volume (i.e. the plane of focus), thereby elegantly avoiding the problem of out-of-focus interference by not generating it in the first place.

Since scattering is lower at the longer wavelengths of excitation used in MP imaging, and also because there is no fluorescence excitation above the plane of focus, there is significantly reduced extinction (relative to confocal microscopy) of the excitation light from regions above the plane of sectioning. In addition, the emission signal does not have to be imaged and is therefore relatively insensitive to scatter. These characteristics provide the deep sectioning capabilities of MP imaging [33]. An additional advantage of MP imaging for *in vivo* studies is that photo-toxic effects are minimized [167]. The benefits of deeper sectioning and improved viability have made MPLSM the tool of choice for developmental biologists who can track fluorescently labeled proteins in both space and time. MPLSM can be used effectively with other techniques such as FLIM and spectral imaging, allowing the scientist to peer deep into the cell and track complicated spatial and temporal processes involving intrinsic (such as NADH) and extrinsic fluorophores (such as GFP-labeled cells). This type of multi-modal experiment presents complex analysis challenges, as multiple dimensions (space, time, lifetime and spectra) can be simultaneously collected [15].

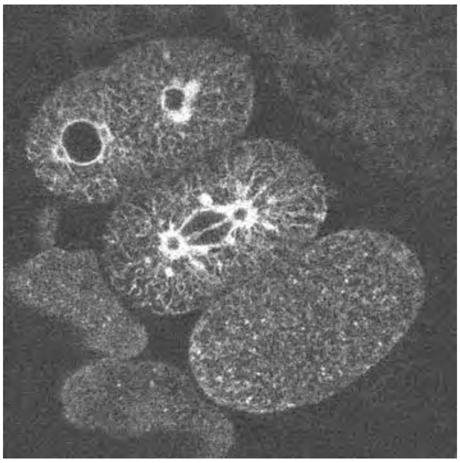


Figure 3 Dynamics of the endoplasmic reticulum in the early nematode embryo. This MP image is of three living *C. elegans* embryos expressing a fusion protein of SP12 (an endoplasmic reticulum protein) and GFP. In these embryos, the endoplasmic reticulum cycles between a highly organized reticulate state during mitosis and a more dispersed state during interphase. The top embryo is a two-cell embryo in which one cell (left) is in interphase and the other cell (right) is entering mitosis. The middle one cell embryo is in the first mitosis, with the endoplasmic reticulum lining the mitotic spindle clearly evident. The bottom one-cell embryo is just post-meiotic and exhibits the more dispersed endoplasmic reticulum organization. (Image courtesy of Dr. Jayne Squirrell, University of Wisconsin–Madison.)

2.1.6 Second Harmonic Imaging

Nonlinear optical effects other than MP fluorescence excitation can occur at the very high photon densities attained at the focus of the scanning excitation beam in a MP microscope (Figure 4). Molecular assemblies with high-order structure, such as collagen matrices, can generate a second harmonic generation (SHG) signal at half the wavelength of the excitation [29]. Unlike MP imaging, the SHG signal has a narrow spectral line-width (determined by the excitation source) and a zero lifetime. These characteristics allow SHG signals to be distinguished from MP signals in a laser scanning microscope with high peak intensity, ultrafast pulse excitation, even if there are fluorescence signals which overlap the SHG signal. These characteristics make SHG imaging a very useful adjunct to MP imaging when observing cells that are embedded in an extracellular matrix. SHG has proven to be effective in investigating processes where collagen plays a critical role such as muscle development [22, 23, 125, 142] and breast cancer progression [26, 148].

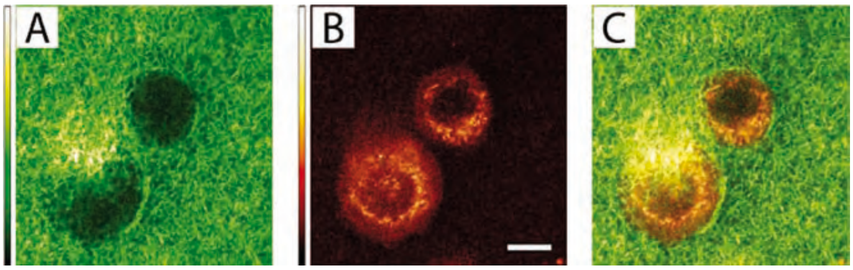


Figure 4 Harmonic imaging. DsRed2-expressing SCC13y cells embedded within a collagen matrix. Collagen second harmonic generation signal (SHG) (A) was collected using a narrow bandpass filter centered at 445 nm and fluorescence from DsRed2 (B) was collected using a 460-nm longpass filter. A 580-nm longpass filter (excluding cellular auto-fluorescence) confirmed that the origin of the fluorescence was exogenous. Panels (A) and (B) are merged in panel (C). This demonstrates the power in live cell imaging of combining endogenous signals such as SHG with fluorescent labels such DsRed. (Image courtesy of Erin Gill, University of Wisconsin–Madison.)

2.2 Functional Genomics

HCS experiments are generally performed to screen through genomic libraries in order to ascertain the function of genes or to screen through compound libraries in order to identify biologically active compounds. Cells can be systematically manipulated either by using RNA interference (RNAi) to interfere with their genes or by exposing them to chemical compounds [124, 153].

2.2.1 RNAi

RNAi is a genetic technique first demonstrated in the nematode *Caenorhabditis elegans* [61, 74] that can be used to block the expression of a target gene by introducing small fragments of double-stranded RNA (dsRNA) with a coding sequence that blocks the gene of interest by binding to its mRNA transcript. This method of DNA silencing has proven to be an extremely powerful tool in *C. elegans* research and has recently been successfully extended to mammalian systems. Four methods have been reported for dsRNA delivery in *C. elegans* [189]: (i) dsRNA injection [61], (ii) feeding with bacteria producing dsRNA [183], (iii) soaking in dsRNA [176] and (iv) *in vivo* production of dsRNA from transgenic promoters [179].

Similarly to *C. elegans*, *Drosophila melanogaster* can also be readily used to study the effects of silencing specific genes using RNAi. The advantage of *D. melanogaster* is that there is a well-developed, cell-based system that can absorb dsRNA directly from solution or when printed on microscope slides using a microarray printer [192]. The combination of microarray technology with this cell-based system allows for an extremely high-density form factor where an entire genome can be assayed on five to 10 standard microscope

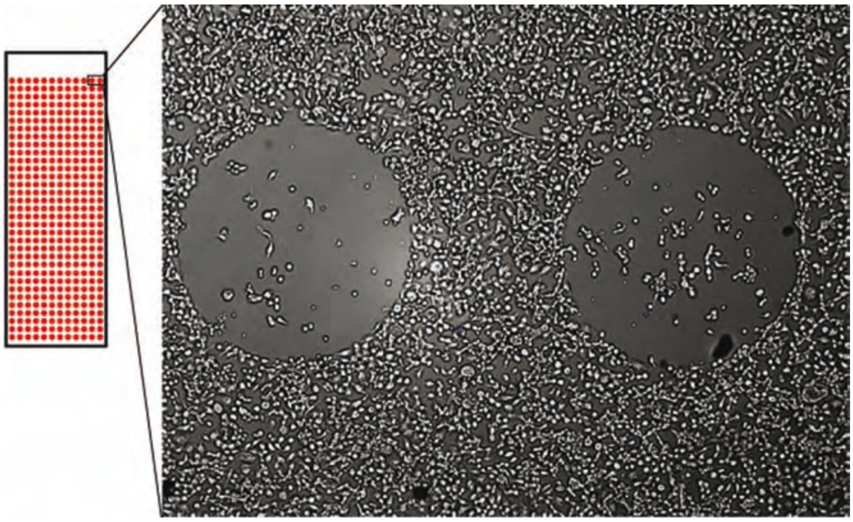


Figure 5 A living cell microarray. On the left is a schematic representation of a standard microscope slide printed with dsRNA "spots" using a standard microarray printer. The spot density can be 2000–5000 per slide depending on the spot size. On the right is a phase-contrast image (×10) of cells growing on top of and around two printed spots of dsRNA that code for a required gene. The phenotype displayed by the cells as a result of "knock-down" of these required genes is lack of growth. Much more subtle phenotypes can be assayed by imaging the cells at higher resolution (×100 is routinely possible), in 3-D, time-lapse, multiple fluorescence channels or a combination of these. As the platform for the assay is a standard microscope slide, 2000–5000 genes can be assayed for one or more phenotypes using a standard fluorescence or confocal microscope with a motorized stage. An organism's entire genome can be represented on 5–10 slides (Figure by Mark Eckley, NIH.)

slides. This platform, called "living cell microarrays", also benefits from using standard microscopes with computer-controlled stages and standard optical equipment instead of the specialized robotics, liquid handling and imaging equipment needed for optical high-density microtiter plates (Figure 5).

High-throughput techniques for RNAi in mammalian cells lag behind those developed for worms and flies. However, several options are under development that share the theme of introducing small interfering RNAs (siRNAs) to cells directly rather than using large dsRNA and allowing the cells to process it. These include digesting a dsRNA library with a nuclease prior to transfection, synthesizing the siRNAs or producing constructs that express siRNA *in vivo* [27,57,75,134,196].

2.2.2 **Chemical Compound Libraries**

Traditionally, chemical compound libraries have been used by the pharmaceutical industry and some academic laboratories to screen compounds for a targeted biological function. With the advent of HCS, the same approaches can now be used for screening a morphological response of the cell to these compounds. The most important aspect of HCS in this application is that knowledge of a molecular target is no longer necessary in order to develop an assay for the compound screen. In HCS, the target of the screen is a morphology that can be reached via potentially many molecular pathways and targets. In this way using HCS to screen chemicals potentially casts a much wider net while simultaneously requiring that compounds are active in cellular context [122, 124].

2.3 Tools for Scientist-driven Analysis Development and Deployment

By necessity, advances in imaging instrumentation have frequently been accompanied by the development of corresponding visualization tools. Although often the visualization needs for these instruments can be met in part by commercial image analysis tools, new instrumentation development can rapidly generate new data types and analysis problems. As the driving image technology is generally not commercially viable, these data types and problems are often beyond the scope of current commercial development, and more flexible, customizable visualization solutions are often needed.

In response to this need there has been a long tradition in the academic microscopy community of the development of freely available image analysis tools that can be harnessed by the microscopist and adapted for their needs. Examples of these imaging tools include tools for 4-D visualization [60, 182], measurement [36] and 3-D reconstruction [42, 109]. Of particular note are tools that are designed to not serve just a specific analysis need, but serve as a general imaging analysis frameworks. Below we discuss two such tools, VisBio and ImageJ, that not only offer a defined set of features, but also a flexible framework to allow additional image analysis techniques to be developed and deployed. Rather than focusing on the development of novel algorithms, these tools are designed to provide flexible environments that can be used to harness existing algorithms.

2.3.1 ImageJ

ImageJ [1] is a public domain image analysis program written in Java, principally by Wayne Rasband of the National Institutes of Health (NIH). Originally designed as a 2-D analysis tool, ImageJ provides image visualization facilities that can be extended via a plugin architecture and by scripting tools accessible to scientists without extensive programming experience. These facilities have encouraged programming contributions from throughout user/developers in the imaging community. Notable user-contributed plugins include the hypervolume browser (http://rsb.info.nih.gov/ij/plugins/hypervolume-browser.html), which can be used to explore 3-D and 4-D data sets, and VolumeJ (http://bij.isi.uu.nl/vr.htm), a plugin for volume rendering. Developers in the ImageJ community have also linked it to other visualization efforts including open-source efforts such as VisAD and the Open Microscopy Environment, and commercial packages such as VTK and MatLab. While ImageJ was not intended as algorithm development project, it has become a highly effective way for scientists to customize their analysis approaches in a way that would often be outside the scope of commercial interests. Like many of the visualization programs discussed below, ImageJ's greatest strength is its user developer group and perhaps more so than any it is designed to be used by the nonprogrammer/developer as well as the experienced scientific programmer.

2.3.2 **VisBio**

The Java-based VisBio (http://www.loci.wisc.edu/visbio) – principally developed by Curtis Rueden at the University of Wisconsin–Madison – is an application for the interactive graphical display and quantitative analysis of biological image data of arbitrary dimensionality [146]. The development of a tool that has support for data of *n*-dimensionality was driven by the emergence of a new class of microscopy data that goes beyond space and time to include new dimensions such as spectra [15, 44, 52, 113] and lifetime [11, 15, 45, 66, 112, 190].

VisBio allows its users to interactively explore and measure the data within 4-D recordings of specimens. In addition to being specially tailored to the demands of handling and animating massive data sets fluidly, VisBio enables the interactive representation of recordings in which each spatiotemporal pixel element contains multiple dimensions, e.g. emission intensity, color spectrum and fluorescence excited state lifetime.

VisBio has been built with the VisAD scientific toolkit [80, 81] – a programming suite that provides capabilities for the analysis and visualization of numerical data (http://www.ssec.wisc.edu/∼billh/visad.html). VisAD's flexible data and display models provide an ideal base from which to design complex applications for visualization and analysis. The data model can be used to express virtually any numerical data, including multispectral images. VisAD's display model enables applications to define arbitrary mappings from numerical variables to various display quantities, such as spatial coordinates, color components, transparency, animation indices, range selectors, contouring, etc. Furthermore, VisAD's support for interaction with displays enables users, for example, to interactively select a pixel in a spatial display and see the corresponding spectrum in another display, and to select a spectral band in the spectral display and see the corresponding image in a spatial display. Other displays map image spatial coordinates to the *x*- and *y*-axes and map spectral band to the *z*-axis, visualized as images stacked up along

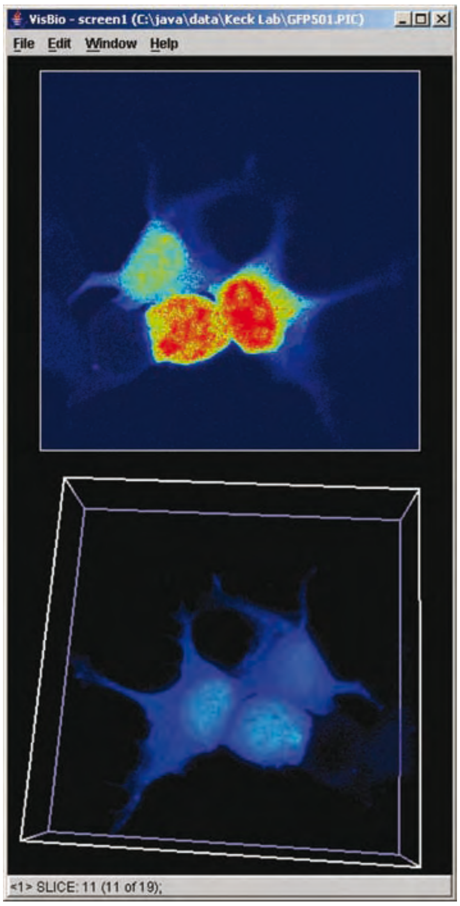

VisBio - screen1 (C:\java\data\Keck Lab\GFP501.PIC)
File
Edit
Window
Help
<1> SLICE: 11 (11 of 19);

Figure 6 Volume Rendering in VisBio 3-D reconstruction of human embryonic kidney cells (HEK293) transfected with a plasmid construct which expresses GFP. The GFP signal is observed throughout the cell. (Confocal data set provided by Dr Carrie Graveel, Mr Lance Rodenkirch and Dr Peggy Farnham of the University of Wisconsin-Madison.)

the z-axis. In some applications, user can interactively draw the outline of a spatial image region, triggering computation of statistics involving only pixels inside the selected region.

VisBio takes advantage of many VisAD features, with the current release providing a general mechanism for handling image data in n dimensions (in fact, it is the only biological imaging software of which we are aware that does so). For example, the software can efficiently browse through a data set consisting of multiple images across time, focal plane and spectral channels, visualizing the images in both 2- and 3-D (Figure 6). The software can understand data of any dimensional organization, including spectral and lifetime dimensions. See Section 4 for further discussion of image analysis software and data management issues.

3 Biological Information Visualization

Familiar types of nonimage biological data share fundamental characteristics that make interpretation and analysis inherently challenging. Annotated sequences, gene expression profiles, pathways, protein interaction maps, phylogenies and other data sets have many data points with complex interactions that are potentially of interest. As these data sets are essentially abstractions that describe underlying physical phenomena, visual representations are not given or necessarily obvious. Strings of nucleotides for sequence data, trees for phylogeny and node–link graphs for pathways are starting points that leave visualization designers with a number of challenges, including scaling up to very large data sets and the appropriate use of available visual cues.

Recent research in the field of information visualization has focused on the development of interactive tools to support the exploration and interpretation of abstract data [164]. This work has involved the application of innovative displays and interaction techniques to data sets from a wide variety of domains [14, 31, 191]. Much of the recent work in bioinformatics visualization builds upon these efforts, particularly with respect to visual encoding of data, multi-scale displays and interaction techniques.

Visual encodings of data define the appearance of individual data items and the relationships between them. Often, the location of items in 2- or 3-D embeddings is of primary importance, encoding either some specific measurements in two or three dimensions (as in a scatter plot) or an indication of relatedness to other items in the data set (as in a graph or network diagram). Size, color coding and shape of representation of data points and connections between them (in graphs) can be used to display values of one or more continuous or categorical attributes. In some cases, redundant coding might be used to provide greater clarity. For data sets involving hierarchical data, appropriate encodings can be used to display containment relationships. Other possibilities include the use of orientation, texture, and animation [31].

Multi-scale displays are a class of strategies designed to address the challenges of interpreting data sets at vastly differing granularities. The classic example of this problem in bioinformatics would be moving from overviews of an entire genome to detailed examination of individual nucleotides. Ideally, visualization systems provide overviews that are compact and usable, detail views that present low-level information in context, and interaction techniques for moving from one extreme to another (or to points in between). Three classes of techniques are commonly used for managing multi-scale displays. Semantic zooming involves modification of the amount of detail displayed to fit the space available: at low magnifications, high-level overviews of large numbers of items can be provided, with detailed information about fewer items at higher magnification [13]. Overview + detail displays provide multiple, linked windows with displays at different scales [163]. Navigation windows in page layout programs are a commonly used overview + detail technique. For data sets involving larger ranges in magnification levels, multiple overviews may be useful. Distortion, or focus + context, techniques warp spatial embeddings to provide proportionately greater space to items of interest, with contextual information displayed around the periphery, forming a "fisheye" view of data [65].

Interaction techniques support browsing, searching and retrieval of details-on-demand. Mouse-over tooltips, hyperlinks, search fields and other fairly common approaches are used in many bioinformatics visualizations. Coordinated multiple views clarify the relationship between differing perspectives on a data set by linking selection across views: when an item is selected in one view, its representation is highlighted in all other views [132]. This coordination can be particularly useful for managing overview + detail displays. Dynamic query tools combine interactive widgets for filtering data sets with fast (below 100 ms) query results and display updates, providing essentially instantaneous feedback [2].

Most tools support some sort of navigation, browsing and searching. Facilities for viewing detail at varying scales of resolution/magnification and retrieving records of interest from related data sets are also common. Other tasks that might be supported include integration of new data sets, statistical analysis, data annotation and construction of scientific arguments. Some tools also provide for the coordination of visualizations of multiple data types and visualization of complex analysis protocols.

A 2001 discussion at the IEEE Information Visualization Conference identified three classes of challenges for bio- and cheminformatics visualization: visual integration of analyses of diverse data sets, high-dimensional analytic visualization and new visualization designs [114]. Although these challenges have been addressed by many of the systems that have appeared in the literature – including those described in this chapter – they are still active problems in need of further research.

3.1 Genome and Sequence Data

With data granularities ranging anywhere from single nucleotides to billions in a sequenced genome, sequence data demonstrates the difficulties of visualization of large data sets. Although tools for examination of synteny at the genome level may not need to provide the nucleotide-level displays needed for analysis of single nucleotide polymorphisms, the ability to move between multiple scales is clearly a useful feature in examining sequence data. Annotations, alignments (local and global) and other task-specific types of data often provide necessary context.

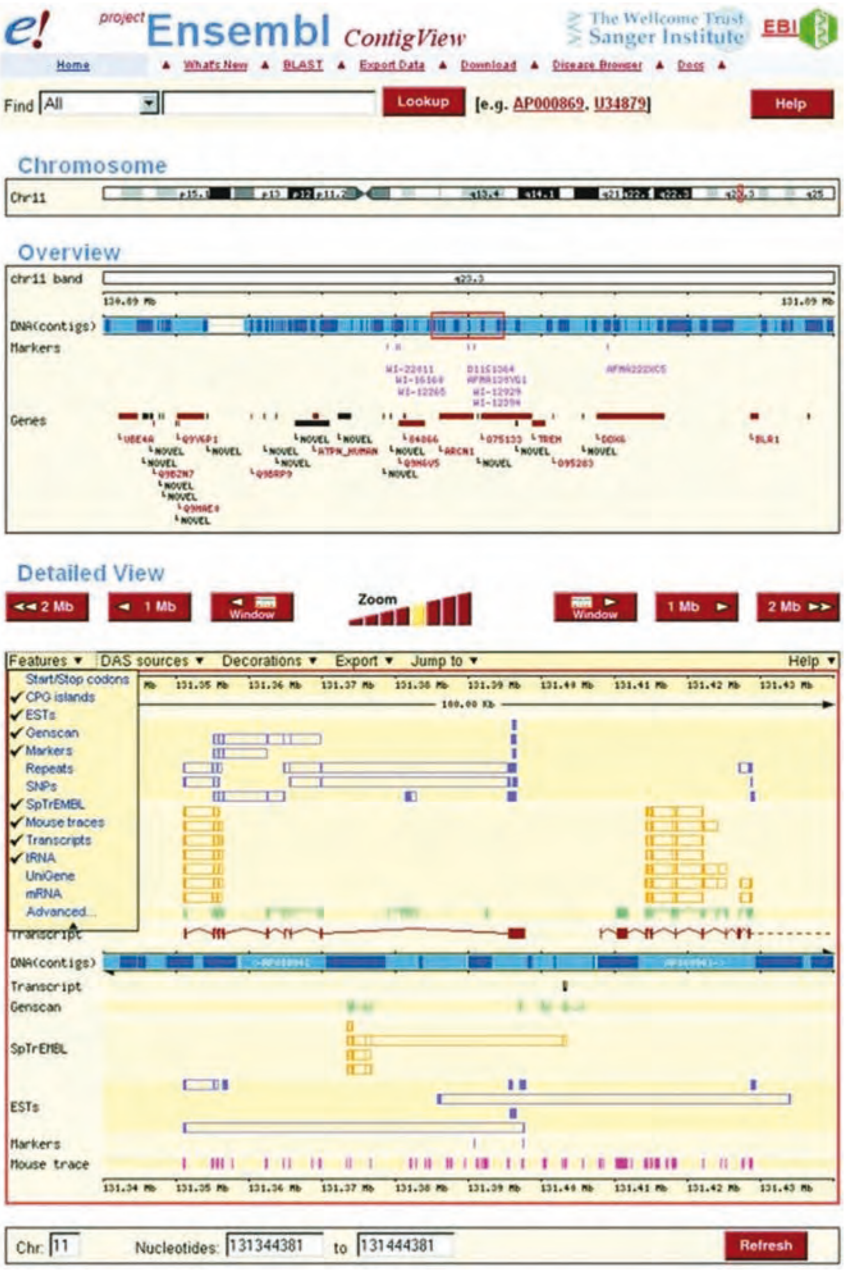
The constant evolution of complete genomes and their annotations makes web interfaces particularly appropriate for genome visualization. Leading web-based sequence visualizations include the University of California at Santa Cruz (UCSC) genome browser (http://www.genome.ucsc.edu) [97] and the Ensembl genome browser (http://www.ensembl.org) [17, 41, 89]. These tools are based on a "track" metaphor: portions of one or more genomes are shown in a horizontal line, with any number of annotations displayed in parallel lines above or below the sequence itself. Information is presented in a dynamically generated image. Users can click on the image to drill-down for more details on any given feature. A desired location in the genome can be accessed via search facilities, by navigating up- or downstream from a given starting point, or by adjusting the magnification. Specific details are given as magnification is increased – a form of semantic zooming. Both tools support integration of tracks from multiple external databases including synteny views for comparative genomics.

Given the wide variety of annotation data sets that might be viewed, configurability is a major concern. Both the Ensembl and UCSC browsers support manual selection of tracks of interest, along with either control over either the decorations used on the tracks (Ensembl) or the density of information on a track (UCSC). Text, graphic and other output formats are provided directly from the browser or via a companion tool [98].

The Ensembl browser's main sequence viewer (ContigView) provides an example of a coordinated, multi-scale view of genomes. A series of views, i.e. entire chromosome, overview, detailed view and base pair view (in the most recent version), are linked together on the same page (Figure 7). Ensembl also provides support for clickable browsing by chromosome location and a per-chromosome synteny view.

Many other web-based track visualization tools have been built. GBrowse, The Generic Genome Browser [169], is a web-based track visualization system designed to generically support multiple model organism databases (MODs).

The track metaphor is also used extensively in sequence applets and stand-alone tools. BioViews [79] was an early effort that used multiscale displays and semantic zooming in an applet environment designed with a specific focus on building reusable widgets. BioViews also provided hyperlinks to external resources, and analysis tools for restriction site mapping and polymerase chain reaction primer design. The Neomorphic GeneViewer annotation tool and ProtAnnot [120] tools are based on several proposed principles for track-based presentations of genome information in stand-along browsers,



Find All

Start/Stop codons
✓ CPG islands
✓ ESTs
✓ Genscan
✓ Markers
Repeats
SNPs
✓ SpTrEMBL
✓ Mouse traces
✓ Transcripts
✓ tRNA
UniGene
mRNA
Advanced...

Chr: 11 Nucleotides: 131344381 to 131444381

Figure 7 The Ensembl genome browser ContigView display, with chromosome, overview and detailed views of a genome (from Ref. [89] by permission of Oxford University Press). The most recent version of ContigView adds a fourth view of the actual base pair sequence.

including continuous, interactive 1-D semantic zooming (as opposed to discrete quantized zooms in response to button clicks), collapsible tracks that can be manually reordered, and the use of color coding and icons to represent alternative transcripts and genome annotations.

Track-based visualizations have also been used for sequence curation. Apollo [117] combines multi-scale track-based views of sequences with drag-and-drop facilities for creating and sequencing editing annotations. Dragging a feature from Apollo's sequence view into an annotation view creates a new gene model, which leads to an annotation for the longest associated open reading frame (ORF). A controlled vocabulary for annotation comments is also provided.

Several tools have adapted the track metaphor to circular displays. The microbial genome viewer is a web-based viewer that displays circular genomes and sequence annotations in concentric circles [101]. GenoMap displays sequence and array expression data on multiple concentric rings, with feature details available on mouse click [152]. CGView [171] produces circular maps that can be zoomed to show individual features (Figure 8). Circular views for noncircular genomes have also been proposed as being useful for identification of relationships between chromosomes [56].

Visualizations based on the embedding of sequences in 3-D space can be useful for comparison and analysis of sequence composition properties. Z-Curves [199] map sequences into curves in 3-D space, with coordinates of the n-th position in the curve determined by the cumulative occurrence counts of the four nucleotides in the first n bases in the sequence. Plotting of multiple curves in a common space supports comparison between sequences.

Sequence alignments present additional visualization opportunities, either as annotation tracks in track-based viewers like the UCSC and Ensembl Genome browsers, or in special-purpose displays. The AlignmentViewer [39] embeds alignment results in a 4-D (3-D and time) space. Any of 12 possible dimensions of alignment data can be mapped to any of the three spatial dimensions or time. When the position of the starting point is included, the alignment is displayed as a 2-D graph. The position of the graph is determined by the values of the remaining spatial axes. The graph extends for the length of the alignment, with values at each position indicating the strength of the alignment. Animation is used to display the time dimension. Two-dimensional range filters can be used to restrict the range of values under consideration for any dimension. Biological Arc Diagrams (BARD [165]) draws arc above (and below, for reverse-strand matches) sequences to link regions with similarities that exceed a given threshold score (Figure 9). Similar figures can also be found in Chapter 8.

A straightforward approach to comparing locations of similar genes or regions in different genes or genomes is to render a single track for each item

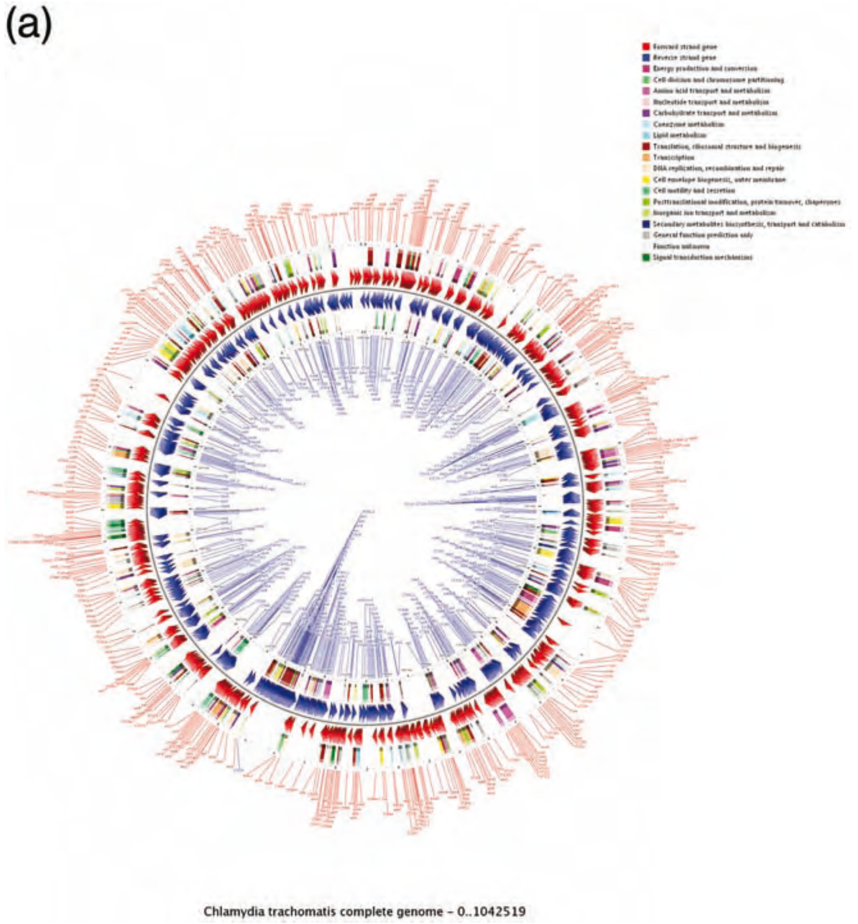

(a)
Forward strand gene
Reverse strand gene
Energy production and conversion
Cell division and chromosome partitioning
Amino acid transport and metabolism
Nucleotide transport and metabolism
Carbohydrate transport and metabolism
Coenzyme metabolism
Lipid metabolism
Translation, ribosomal structure and biogenesis
Transcription
DNA replication, recombination and repair
Cell envelope biogenesis, outer membrane
Cell motility and secretion
Posttranslational modification, protein turnover, chaperones
Inorganic ion transport and metabolism
Secondary metabolites biosynthesis, transport and catabolism
General function prediction only
Function unknown
Signal transduction mechanisms
Chlamydia trachomatis complete genome – 0..1042519

Figure 8a CGView output: full zoomed views of *Chlamydia trachomatis* genome [171]. (From http://wishart.biology.ualberta.ca/cgview/gallery.html, used by permission.)

being compared, along with lines or other glyphs going between the tracks to connect similar regions. This approach is used in the GenomePixelizer [106] and Apollo [41].

The PIP (percent identity plot) displays homology by showing the position in one sequence and the similarity (percentage) for each aligning segment from the second sequence [155]. The MultiPip extends this idea to alignments of multiple sequences [154]. A MultiPip contains individual pips comparing sequences against a common reference sequence. Features including genes, exons, and repeats can be displayed within each Pip (Figure 10).

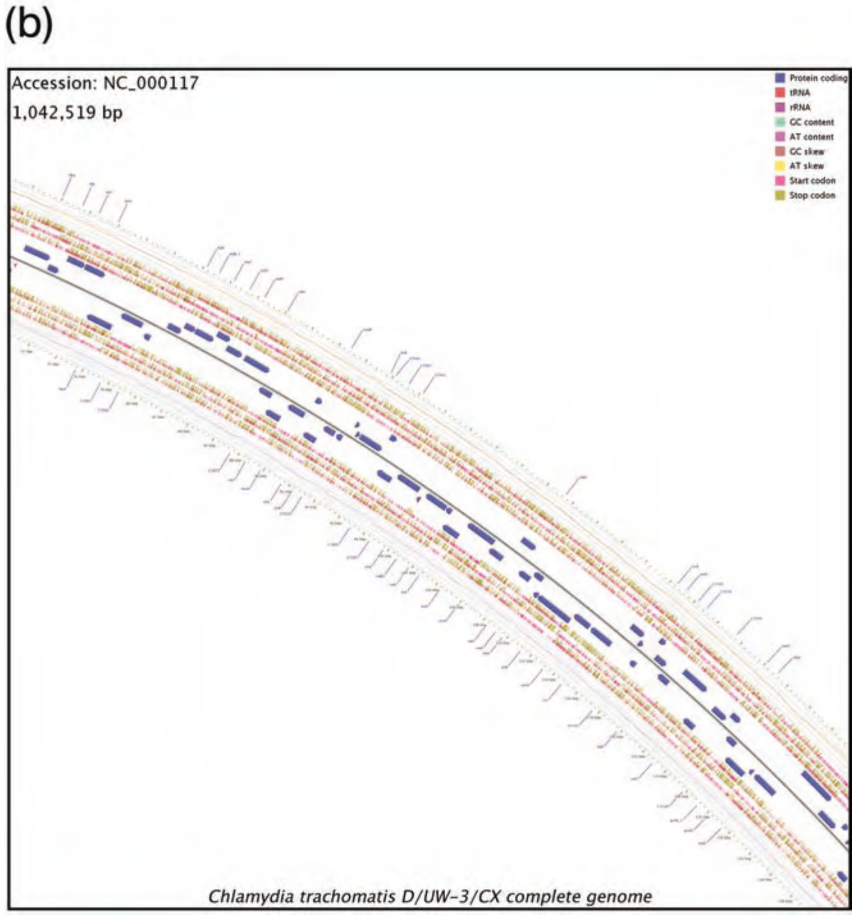


Figure 8b CGView output: Zoomed views of *Chlamydia trachomatis* genome [171]. (From http://wishart.biology.ualberta.ca/cgview/gallery.html, used by permission.)

VISTA and related tools use similar identity plots to display global alignment results. VISTA Browser [64] is a web browser that displays percentage similarities for two genomes. Similar regions are highlighted and color coded for coding and noncoding, with zooming, drill-down and search facilities. Given a multiple alignment and an associated phylogenetic tree Phylo-VISTA [159] generates similarity ratings for each internal node in the tree. Thus, similarities are defined relative to consensus sequences, as opposed to defining one of the input sequences as a reference point. An interactive display of the phylogenetic tree can be used to select the internal nodes that are displayed.

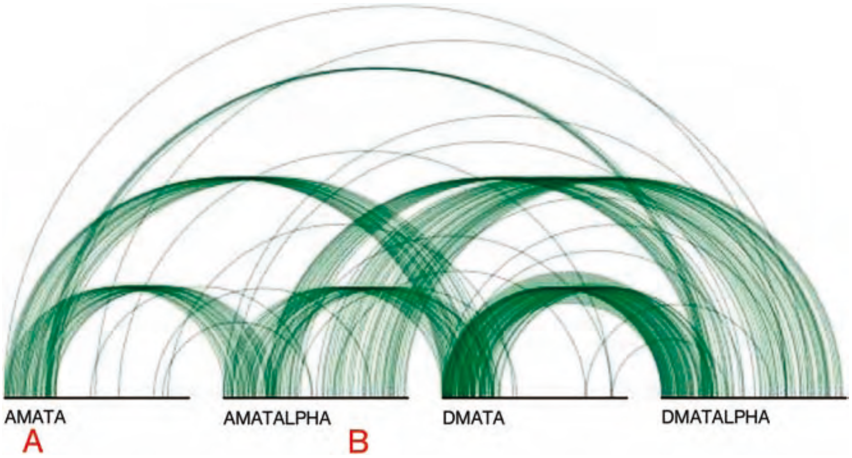


Figure 9 A BARD view using arcs to link similar sequences in two strains of *Cryptococcus.* (From Ref [165], © 2003 IEEE, used by permission).

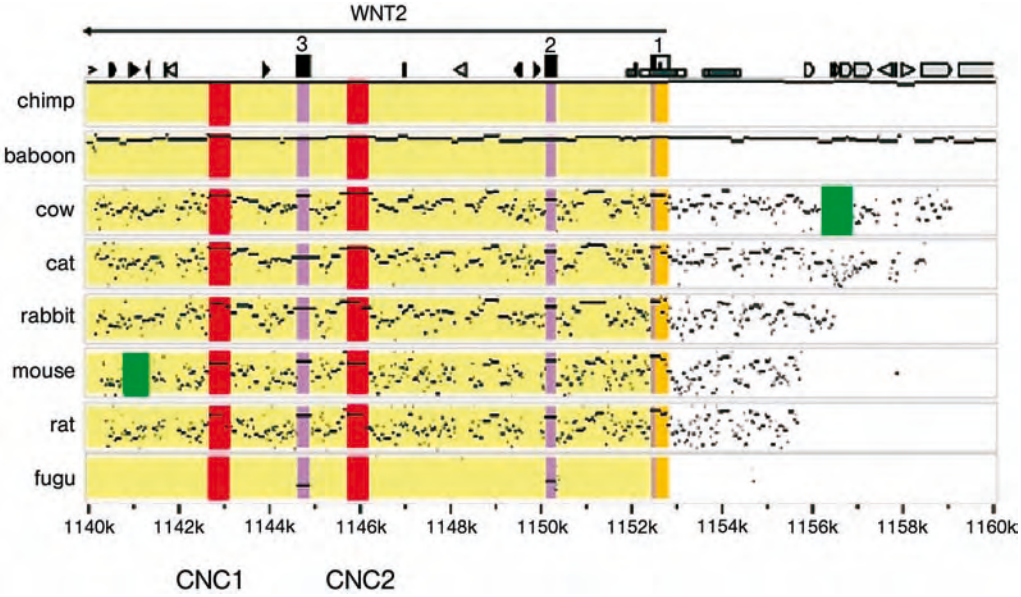


Figure 10 MultiPip of the *WNT2* region. Local alignments between the human genome and each of eight other genomes are plotted. For each genome, the percentage identity of each gap-free alignment segment is plotted. Annotations above the plots indicate features in the human genomes. Color coding in the plots indicates introns, exons, noncoding regions and deletions. (From Ref. [154], by permission of Oxford University Press.)

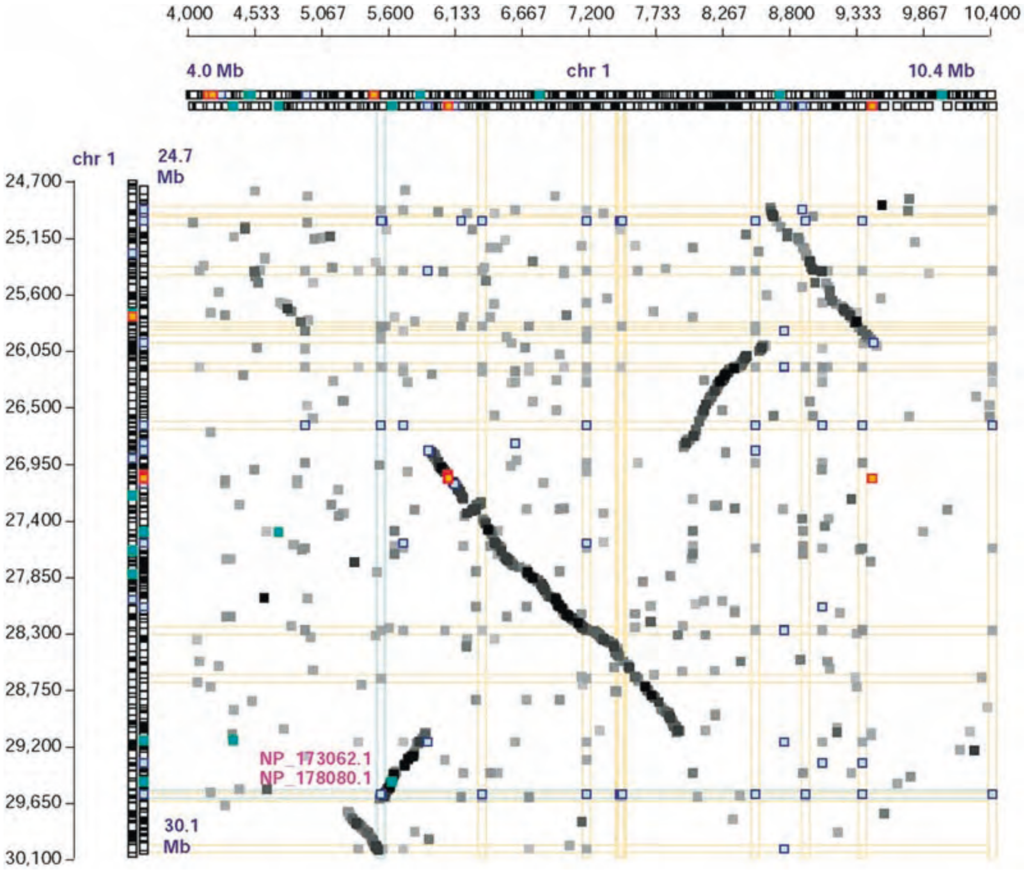


Figure 11 A GenoPix2D dot plot of *Arabadopsis* chromosome 1. Three gene families are color coded in the axes. Vertical and horizontal lines highlight homologs to the genes corresponding to the selected dot. (From Ref. [30], used by permission).

Other approaches to displaying large-scale homologies have used different perspectives include GenoPix2D [30], an interactive dot plot, which supports querying, filtering, zooming and coloring of genes by family membership (Figure 11), and DisplayMUMS, which shows alignments of individual shotgun sequences to a global alignment [108].

Noting the possibilities of a 3-D view, the Sockeye browser displays multiple genomes as parallel tracks in a projected 3-D space, using the height of annotations to display values such as alignment scores and alignment site confidence predictions [126]. As these scores are also color- coded, the height coding is redundant.

A variety of tools have been created for visualizing more specific types and applications of sequence data. viewGene [100] and GeneWindow [168] are track-based web browsers for SNP data. Gbuilder is a Java tool for visualization of EST clusters [128], while ESTminer provides similar functionality in a web interface [88]. ESTviewer displays ESTs that are conserved in the human genome and alternatively-spliced variants [35]. The Maize Mapping Project's iMap Viewer [58] provides side-by-side displays of genetic and physical maps of the maize genome, complete with links to external data. See Table 1 for selected genome visualizations.

Table 1 Selected genome visualizations

System type	Examples
Web-based browser	UCSC Genome Browser [97], Ensembl [89]
Stand-alone browsing and annotation	Neomorphic GeneViewer/ProtAnnot [120], Apollo [117]
Circular display	Microbial Genome Viewer [101], GenoMap [152], CGView [171], ChromoWheel [56]
Local alignments	Alignment Viewer [39], Biological Arc Diagram [165]
Large-scale homology	GenomePixelizer [106], GenoPix2D [30], PipMaker [155], MultiPipMaker [154], VISTA [64], Phylo-VISTA [159], DisplayMUMS [108], Sockeye [126]

3.2 Gene Expression Data

Gene expression data sets from microarray experiments can involve measurements for thousands of genes in dozens of cell conditions or cell states (called treatments from now on). The interpretation challenge generally involves identifying sets of genes with similar profiles, individual genes with distinctive profiles and, possibly, samples with profiles that might indicate the presence of a regulatory relationship (see also Chapters 24–27).

Classic approaches to visualizing microarray data are based on similarity clustering of profiles.

Red–green expression maps involve a combination of hierarchical clustering represented by a dendrogram and a dense, color-coded display called a heat map. Genes are clustered by calculating pairwise measures, replacing the two most similar samples with a cluster represented by their average and continuing until only one cluster remains. The genes are then ordered and displayed in a table, with one row for each gene and one column for each treatment. Each cell in the table contains a color-coded expression value, with red indicating increased expression and green indicating decreased expression [55] (Figure 12).

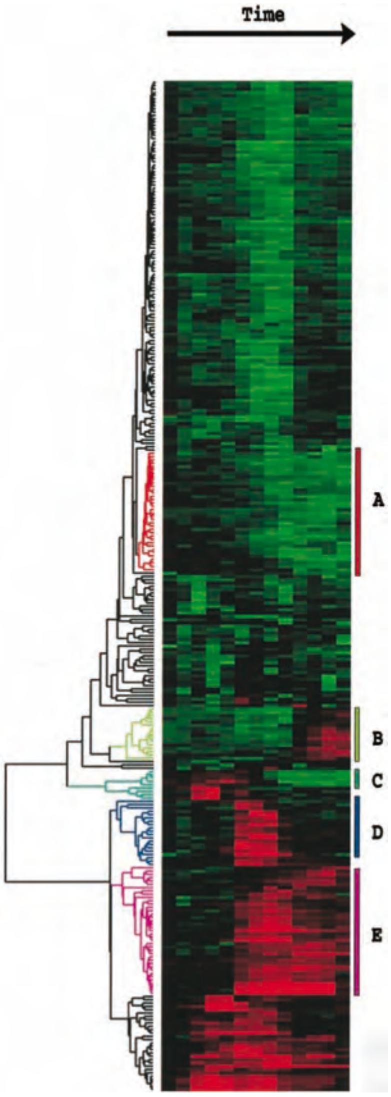


Figure 12 Gene expression values for a microarray time-course experiment. Each row is a gene, and each column is a time point. Cells are color coded to indicate expression levels relative to time 0: green readings are repressed, and red enhanced, with saturation level indicating the magnitude of the change. The dendrogram on the left groups the genes into clusters, with the lengths of the branches indicating levels of similarity. Five clusters of interest are marked on the right. (From Ref. [55], © 1998, National Academy of Sciences USA, used by permission.)

A wide variety of other clustering and dimensionality reduction techniques have been applied to microarray data sets. Techniques including self-organizing maps (SOMs) [105,177,184], multidimensional scaling [6,200], self-adaptive networks [188] and graph-theoretic methods [161] have been used to generate graphical maps indicating relationships between individual genes and clusters of genes. The MultiExperimentViewer (http://www.tm4.org/mev.html) supports comparative visualization of clusters generated by many of these algorithms.

Interactive array visualization tools provide user controls built upon the (generally) static maps created by clustering algorithms. Examples include animated SOMs to illustrate gene expression changes over multiple time points [54], and navigable interactive plots of clustered results [162]. GeneXplorer [144] provides linked dendrograms at multiple resolutions in a web-based application. Java Treeview [149] extends Eisen's original work with a multiscale view of a dendrogram, scatterplots, karyoscopes and links to external web resources. VistaClara [102] takes a different approach to extending basic heat maps, providing facilities for adding annotations, the ability to reorder rows (and columns, for comparisons of multiple experiments) based on similarity measures and an alternative rendering scheme that uses differentially sized ink blots to indicate differences in expression values. Genesis [173] is a Java tool that supports multiple clustering approaches and multiple views – including dendrograms and 3-D projection plots. The Hierarchical Clustering Explorer (HCE) [158] adds interactivity to dendrograms, including dynamic query controls for selecting an appropriate number of clusters, coordinated displays linking dendrogram views to 2-D scatter plots of arbitrary dimensions and linked views for comparison of alternative clustering algorithms (Figure 13). A version of the HCE for use in microarray probe design has also been developed [157].

Many of these techniques for generating and displaying clusters suffer from potentially misleading artifacts associated with the choices of color saturation levels and display reference points. GeneVAnD supports coordinated views of rank-based grayscale coloring of genes, views that indicate differences between genes and cluster averages, and interactive principal components analysis (PCA) projections [82].

Commercial products for visualizing microarray data include Spotfire Decision Site (http://www.sptofire.com), Agilent's GeneSpring (http://www.agilent.com), and OmniViz (http://www.omniviz.com). These products include support for data analysis and clustering along with customizable 2- and 3-D scatter plots and color-coded expression plots. Spotfire and GeneSpring also provide support for workgroups and relational database storage of expression data.

Clustering-based approaches are useful for large-scale analysis of microarray data sets. An alternative approach is the use of queries to search for patterns of interest in subsets of the data. For time-course arrays, these searches might involve, for example, the search for transcriptional targets via

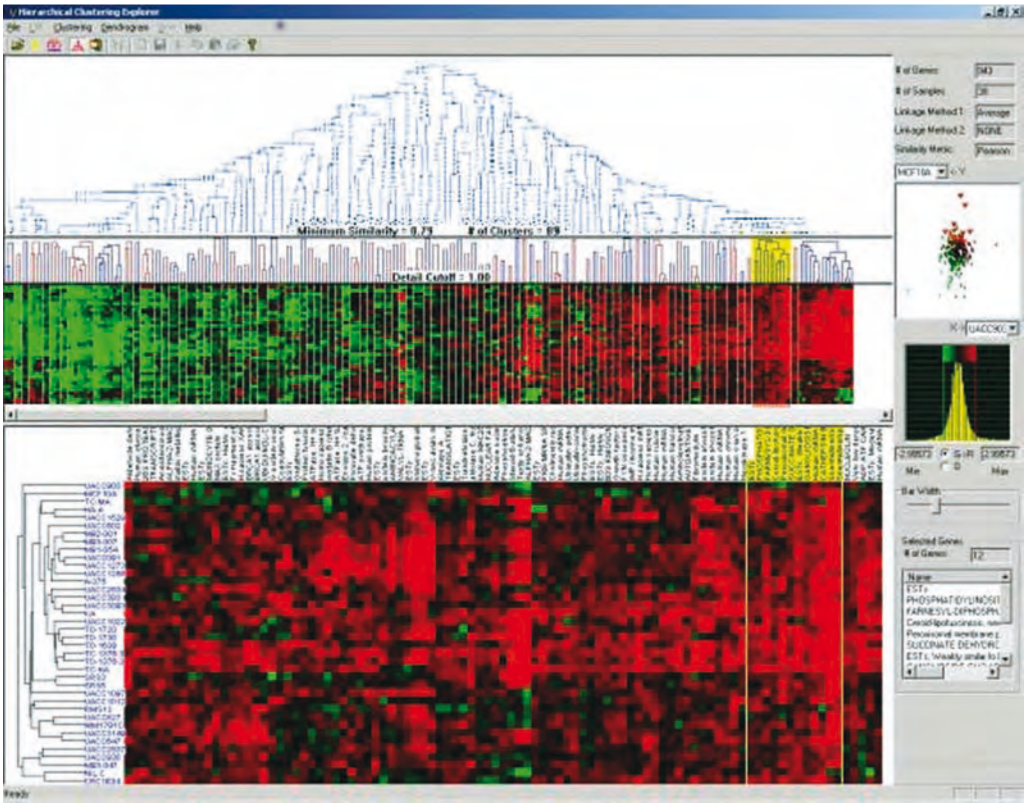


Figure 13 The HCE. An overview of a microarray data set is shown in the top pane. The "minimum similarity bar" has been dragged to split the dendrogram into multiple clusters. Details for the selected cluster (highlighted in yellow) are shown in the bottom pane. A scatter plot of items in that cluster is shown on the right. (From Ref. [158], © 2003 IEEE, used by permission).

the identification of genes with expression patterns that change significantly after changes in a known transcription factor. TimeSearcher is a general time series analysis tool that uses rectangular regions drawn on time series plots as queries, restricting results to items in a data set that have values in a given value range during specified times. This approach has been used to find regulatory targets for known transcription factors [83, 84] (Figure 14). The Time-series Explorer presents an alternative view, using a scatter plot to compare the change from normal expression over a selected interval (y-axis) to the change in values from the beginning of the interval to its end (x-axis), with animation conveying the changes in the interval [47].

Visual tools that place gene expression data in the context of annotations from the Gene Ontology (GO) [4] (see also Chapter 29) can be useful for identi-

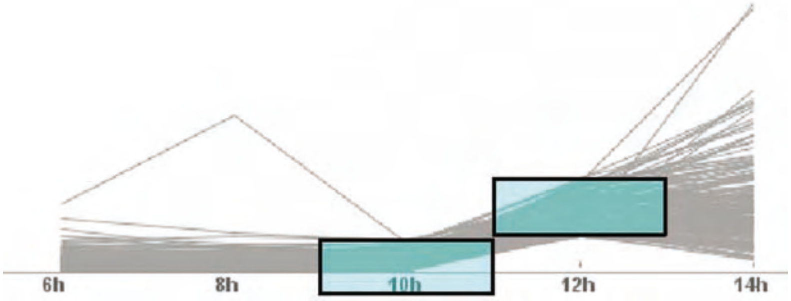
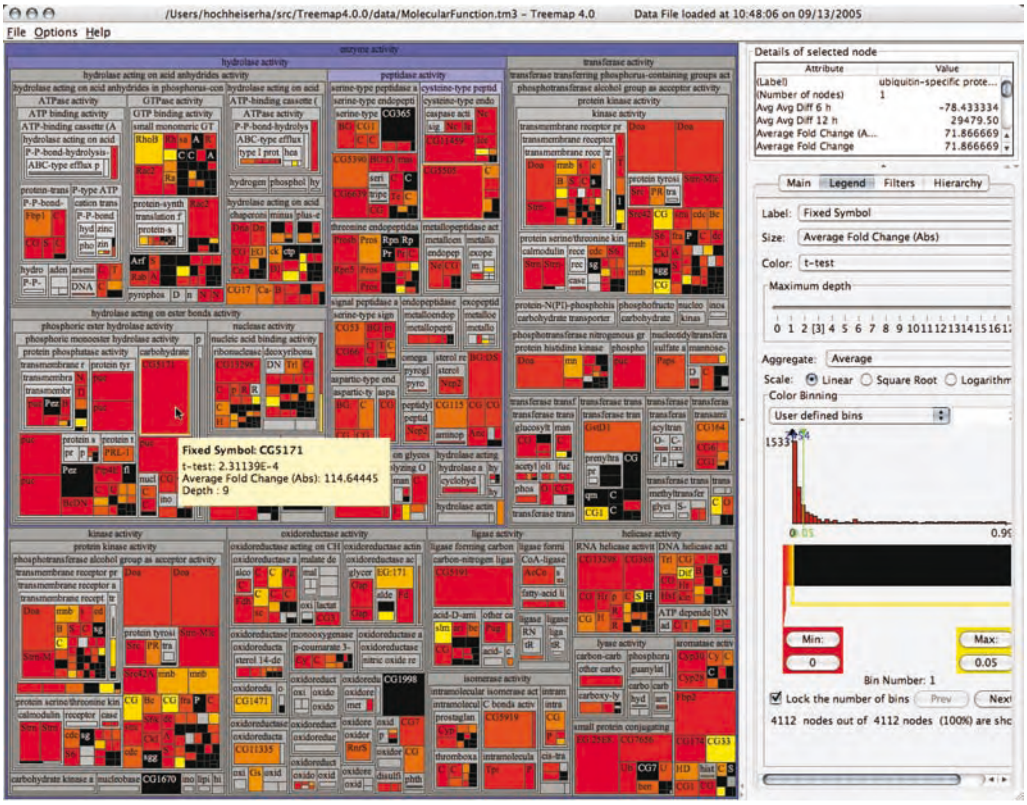


Figure 14 TimeSearcher queries specifying genes with low expression levels at the 10-h time point and higher levels at 12 h. (From Ref. [84], reproduced with permission of Palgrave Macmillan.).

fying groups of genes with similar roles that have similar expression patterns. GoSurfer is a Windows-based program that displays any one of the three GO hierarchies in a traditional tree view, with branch color coding based on relative expression levels [201]. Similar functionality is provided by GoMiner, which displays scalar vector graphics (http://www.w3.org/Graphics/SVG) trees with expression-level color coding alongside traditional tree-based browsers of the GO tree, annotated with expression levels [198]. Treemap is a space-filling hierarchy visualization. Given a hierarchy and a rectangular space, Treemap recursively divides the space based on the size of the contained nodes. Given an array data set, a Treemap display can use the relative change in expression level to size each gene, and the significance value for that gene as a color code, thus providing a concise view of GO categories with highly expressed genes of differing significance [8] (Figure 15). GenePlace and TreeMapClusterView [123] provide alternative uses of treemaps for GO and microarray data.

Another class of visualization tools addresses the challenges of interpreting comparative genomic hybridization (CGH) results. CGH uses microarray analysis of complementary DNA of multiple genomes to identify regions with sequence deletions or duplications [138]. These deletions and duplications lead to changes "copy number": deviations from expected DNA levels for specific regions. As interpretation of these results often involves examinations of copy numbers in a genomic context, visualization of CGH data often involves displays of results alongside genomic maps. SeeGH [38] and CGHPRO [37] combine displays of ratio information with individual chromosome maps with an individual chromosome view and a variety of parameterized filters.

VistaChrom [103] extends this approach with additional views at the level of individual genes and probes, with coordinated highlighting in the four



Data File loaded at 10:48:06 on 09/13/2005

Fixed Symbol: CG5171
t-test: 2.31139E-4
Average Fold Change (Abs): 114.64445
Depth : 9

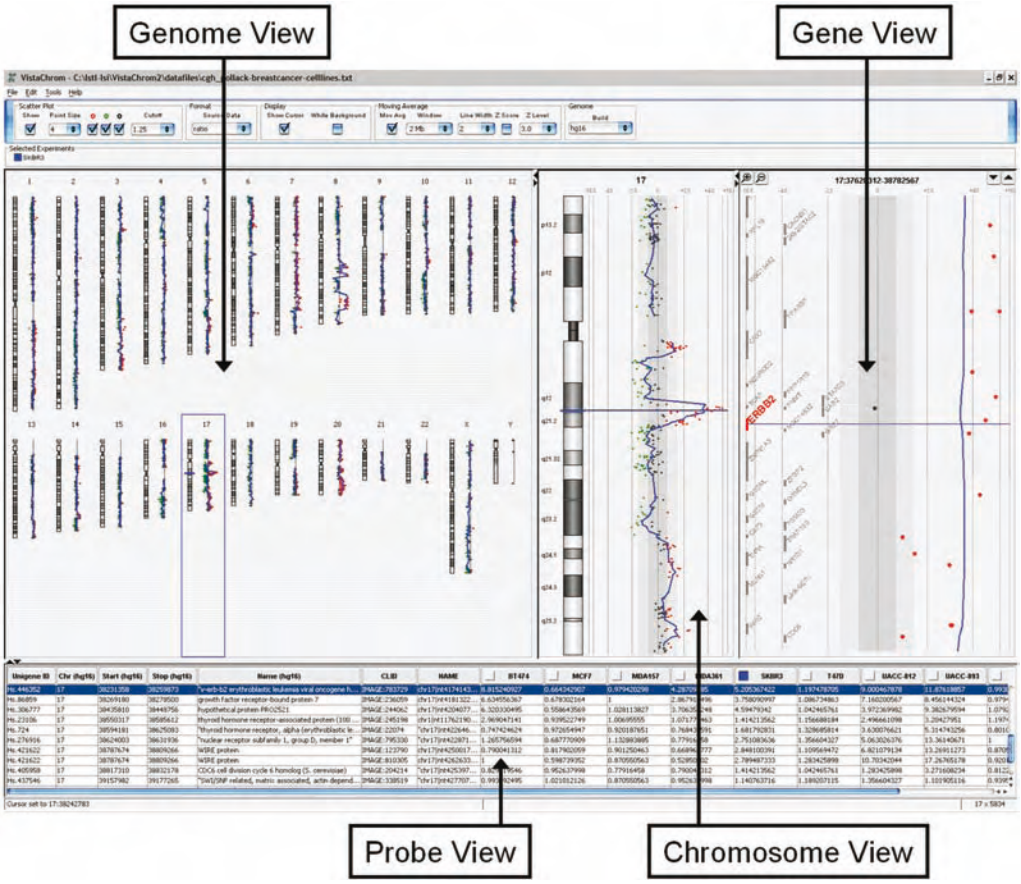
Details of selected node

Attribute	Value
(Label)	ubiquitin-specific prote...
(Number of nodes)	1
Avg Avg Diff 6 h	-78.433334
Avg Avg Diff 12 h	29479.50
Average Fold Change (A...	71.866669
Average Fold Change	71.866669

Label: Fixed Symbol
Size: Average Fold Change (Abs)
Color: t-test
Maximum depth
0 1 2 [3] 4 5 6 7 8 9 10 11 12 13 14 15 16 1
Aggregate: Average
Scale: ● Linear ○ Square Root ○ Logarithm
Color Binning
User defined bins
1533
0 0.05 0.99
Min: 0
Max: 0.05
Bin Number: 1
☑ Lock the number of bins
4112 nodes out of 4112 nodes (100%) are sho

Figure 15 Treemap view of gene ontology, coded with expression values for a microarray data set. The rectangular space is recursively divided into subregions for each of the children of a given GO node. Individual genes are labeled with gene names, with the size of each gene corresponding to the expression level and color indicating significance level (black being insignificant). (From Ref. [8], used by permission.)

views (Figure 16). VistaChrom also provides additional statistical views, a summary "aberration view" and support for analysis of multi-array experiments. Caryoscope [5] provides genomic overviews without actual chromosome maps, with novel zooming techniques supporting the visualization of individual features in context. CGHAnalyzer [72] provides views at the chromosome level, with a focus on analysis of multiple experiments, with companion tools CircleViewer and CGHBrowser, respectively, providing a circular overview of a single experiment and high-resolution graphs of raw array data. ChARMView [130] combines visualization of Array CGH and gene expression array data from multiple experiments with a novel analysis algorithm.



Unigene ID	Chr (hg16)	Start (hg16)	Stop (hg16)	Name (hg16)	CLID	NAME	BT474	MCF7	MDA157	MDA361	SKBR3	T47D	UACC-812	UACC-893	
Hs.446352	17	38231358	38259873	"v-erb-b2 erythroblastic leukemia viral oncogene h...	IMAGE:783729	chr17|nt4174143...	8.815240927	0.664342907	0.979420298	4.2870[illegible]95	5.205367422	1.197478705	9.000467878	11.87618857	0.993
Hs.86859	17	38269180	38278500	growth factor receptor-bound protein 7	IMAGE:236059	chr17|nt4181322...	6.634556367	0.678302164	1	2.8679[illegible]496	3.758090997	1.086734863	7.160200567	8.456144324	0.979
Hs.306777	17	38435810	38448756	hypothetical protein PRO2521	IMAGE:244062	chr17|nt4204077...	6.320330495	0.558643569	1.028113827	3.7063[illegible]248	4.59479342	1.042465761	3.972369982	9.382679594	1.079
Hs.23106	17	38550317	38585612	thyroid hormone receptor-associated protein (100 ...	IMAGE:245198	chr1|nt11762190...	2.969047141	0.939522749	1.00695555	1.0717[illegible]463	1.414213562	1.156688184	2.496661098	3.2042795	1.197
Hs.724	17	38594181	38625083	"thyroid hormone receptor, alpha (erythroblastic le...	IMAGE:22074	"chr17|nt422646...	0.747424624	0.972654947	0.920187651	0.7684[illegible]591	1.681792831	1.328685814	3.630076621	5.314743256	0.801
Hs.276916	17	38624003	38631936	"nuclear receptor subfamily 1, group D, member 1"	IMAGE:795330	"chr17|nt422871...	1.265756594	0.687770909	1.132883885	0.7791[illegible]58	2.751083636	1.356604327	5.063026376	13.36140671	1
Hs.421622	17	38787674	38809266	WIRE protein	IMAGE:123790	chr17|nt4250017...	0.790041312	0.817902059	0.901250463	0.6689[illegible]777	2.848100391	1.109569472	6.821079134	13.26911273	0.870
Hs.421622	17	38787674	38809266	WIRE protein	IMAGE:810305	chr17|nt4262633...	1	0.598739352	0.870550563	0.5295[illegible]02	2.789487333	1.283425898	10.70342044	17.26765178	0.920
Hs.405958	17	38817310	38832178	CDC6 cell division cycle 6 homolog (S. cerevisiae)	IMAGE:204214	"chr17|nt425397...	0.82[illegible]9546	0.952637998	0.77916458	0.7900[illegible]312	1.414213562	1.042465761	1.283425898	3.271608234	0.812
Hs.437546	17	39157982	39177265	"SWI/SNF related, matrix associated, actin depend...	IMAGE:338519	"chr17|nt427707...	0.99[illegible]92495	1.021012126	0.870550563	0.9526[illegible]998	1.140763716	1.189207115	1.356604327	1.101905116	0.939

Cursor set to 17:38242783

17 x 5834

Figure 16 VistaChrom visualization of array CGH data. In the genome view, copy number variations are shown alongside cytogenetic maps of each chromosome. Progressively greater levels of detail are provided in the chromosome view and gene view, with full details in the probe view. Linked interactions in all windows provide contextual information. (Figure courtesy of Robert Kincaid [103].)

A variety of more specific microarray visualization tools have been developed for other data types. Examples include tools for data sets involving circular genomes [101, 152] and SNPs [180]. ExpressionView combines microarray data with quantitative trait loci (QTL) information, showing loci from different samples alongside chromosome maps [62]. Other tools overlay gene expression level indicators on top of genetic maps [9] or microscopy images [67]. See Table 2 for selected visualizations of gene expression data.

Table 2 Selected visualizations of gene expression data

System type	Examples
Interactive cluster analysis	GEDI [54], NIA Array Analysis Tool [162], GeneXplorer [144], Java TreeView [149], VistaClara [102], Genesis [173], HCE [156, 158], GeneVAnD [82], Spotfire DecisionSite (http://ww.spotfire.com), Agilent GeneSpring (http://www.agilent.com)
Time-series microarray analysis	TimeSearcher [84], Time-Series Explorer [47]
Microarray and gene ontology annotation	GoSurfer [201], GoMiner [198], Treemap [8], Gene-Place/TreeMapClusterView [123]
ArrayCGH	SeeGH [38], CGHPRO [37], VistaChrom [103], Caryoscope [5], CGHAnalyzer [72], ChARMView [130]

3.3 Proteomics

Proteomics tools provides visualizations of proteomic information and protein interaction data in a genomic context. The UCSC Proteome Browser [86] displays protein information in a series of tracks, similar to those used in genome browsers. Amino acid sequences are displayed along with tracks for genomic sequences, exons, polarity, and other details. A variety of genome-wide histograms are also available, along with external links to related resources. PQuad [76] provides a varying perspective on the use of tracks for protein data, using a "wrapped line" metaphor to display long sequences on multiple lines that wrap like text on a page. PQuad provides linked views at three scales (overviews, individual ORFs and peptide sequences), along with filters, displays of predicted protein functions and difference visualizations for comparing the output of multiple experiments.

Visualization of protein families can provide perspective on relationships between proteins. TreeWiz [145] provides a tree visualization of more than 70 000 clustered proteins. Zooming and filters for subtrees are supported.

3.4 Interaction Networks and Pathways

Understanding the relationships between biological entities is clearly a core challenge of bioinformatics. Visual depictions of interaction networks and biological pathways are widely used to interpret this data. These visualizations are generally drawn as graphs, with proteins, genes or biological states as nodes and edges connecting interacting nodes. The use of color coding to indicate interaction confidence and GO annotations (e.g. Ref. [68]) can aid interpretation, but these large, dense graphs with thousands of interactions between thousands of proteins present significant challenges for visualization and interpretation. Interactive tools (both standalones and applets) for network visualization generally combine graph layouts with search facilities, zoom and pan, and filters based on confidence levels [46], GO annotations [34], and other criteria.

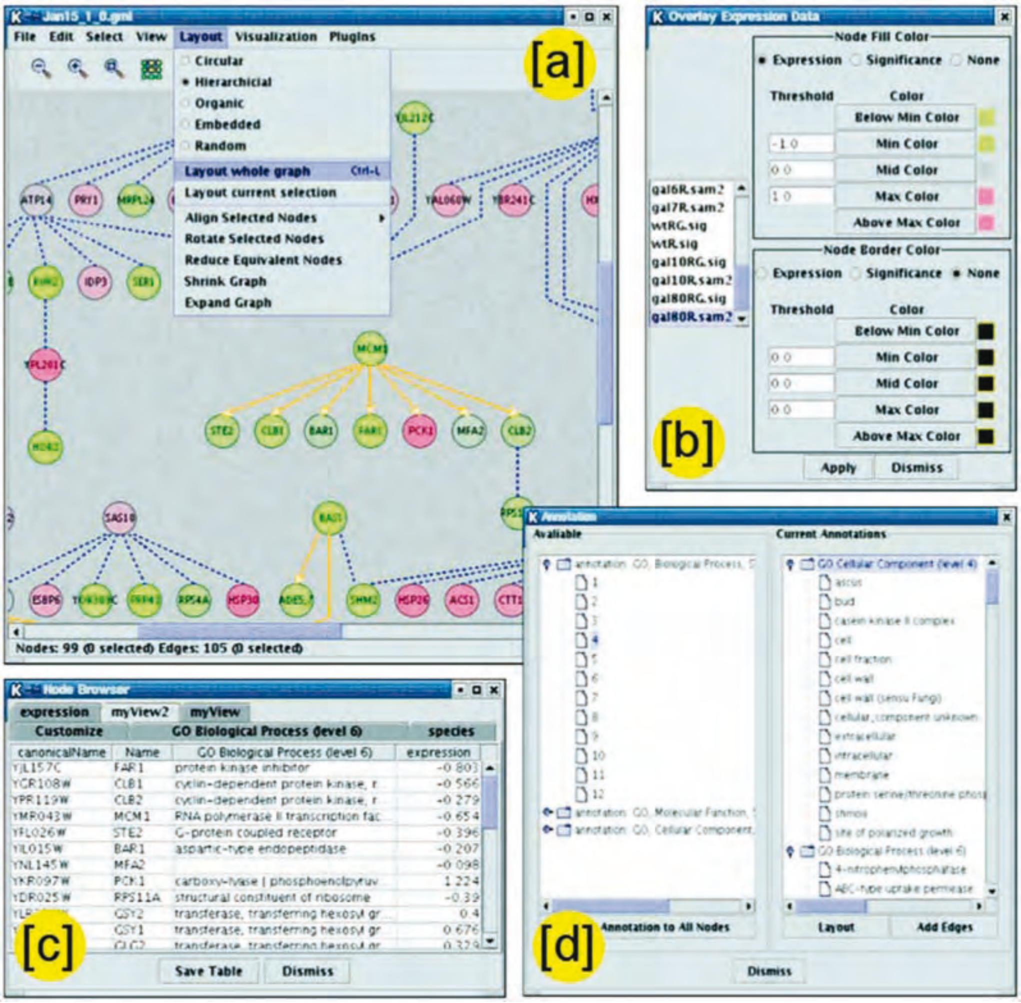
Large interaction networks generated from protein–protein interaction investigations (e.g. Refs. [63,68]) and similar experiments can contain thousands of nodes and a dense interaction structure. This complexity can be tamed somewhat via techniques that identify potentially promising subgraphs. Possibilities include graph-theoretic analysis of network structure [28], and integration of interaction results from differing sources along with genetic and gene expression data [7].

VisANT [87] is a web-based, database-driven interaction visualization system that integrates data from a wide variety of sources, including pathway and homology data. Osprey [25] combines color coding of interaction edges based on the experimental system with the use of GO categories to color-code nodes in interaction diagrams. Both types can indicate multiple values: interactions from multiple systems are segmented into multiple colors, while multi-colored pie charts are used for nodes with multiple GO categories. ProViz [93] adds support for managing views of multiple subgraphs of a network.

Cytoscape [160] is an extensible platform for integrating interaction networks with additional data sources, including expression profiles, phenotypes, and databases of functional annotations (Figure 17). Cytoscape also provides a plugin architecture that has been used to build bridges to the Systems Biology Workbench [90] and the Biomolecular Interaction Network Database [3].

Biological pathway diagrams are powerful tools for synthesizing experimental results into models that describe complex cellular behaviors. Often pathways are created by hand, and are therefore generally smaller than interaction diagrams. Interactions in these diagrams often have associated annotations, which are often visualized as edge decorations. Web-accessible pathways databases, such as EcoCyc [99] and KEGG [96], combine searching or browsing for pathways of interest with clickable pathway diagrams that provide drill-down access to specific detail (see also Chapter 20). Interactive pathway tools typically support both pathway creation and visualization. For example, the Genomic Object Net (GON) provides tools that can be used to both edit KEGG and other pathways and to create customized visualizations of simulations [131].

Despite efforts to define a visual language for pathway representation [115], pathway tools use differing techniques to encode information. Often the appearance of nodes in links is customizable [85,160], although some tools use customized glyphs [50]. Frequently, search facilities are provided for limiting the display to items of interest. GSCope [185] augments search facilities with



Nodes: 99 (0 selected) Edges: 105 (0 selected)

Node Fill Color

● Expression ○ Significance ○ None

Threshold	Color
	Below Min Color 🟨
-1.0	Min Color 🟨
0.0	Mid Color
1.0	Max Color 🟥
	Above Max Color 🟥

gal6R.sam2
gal7R.sam2
wtRG.sig
wtR.sig
gal10RG.sig
gal10R.sam2
gal80RG.sig
gal80R.sam2

Node Border Color

○ Expression ○ Significance ● None

Threshold	Color
	Below Min Color ■
0.0	Min Color ■
0.0	Mid Color ■
0.0	Max Color ■
	Above Max Color ■

Apply Dismiss

expression myView2 myView

Customize | GO Biological Process (level 6) | species

canonicalName	Name	GO Biological Process (level 6)	expression
YJL157C	FAR1	protein kinase inhibitor	-0.803
YGR108W	CLB1	cyclin-dependent protein kinase, r...	-0.566
YPR119W	CLB2	cyclin-dependent protein kinase, r...	-0.279
YMR043W	MCM1	RNA polymerase II transcription fac...	-0.654
YFL026W	STE2	G-protein coupled receptor	-0.396
YIL015W	BAR1	aspartic-type endopeptidase	-0.207
YNL145W	MFA2		-0.098
YKR097W	PCK1	carboxy-lyase | phosphoenolpyruv...	1.224
YDR025W	RPS11A	structural constituent of ribosome	-0.39
YLR[illegible]	GSY2	transferase, transferring hexosyl gr...	0.4
[illegible]	GSY1	transferase, transferring hexosyl gr...	0.676
[illegible]	GLG2	transferase, transferring hexosyl gr...	0.329

Save Table Dismiss

Available | Current Annotations

annotation: GO, Biological Process, 5
1
2
3
4
5
6
7
8
9
10
11
12
annotation: GO, Molecular Function, 5
annotation: GO, Cellular Component, 5

GO Cellular Component (level 4)
ascus
bud
casein kinase II complex
cell
cell fraction
cell wall
cell wall (sensu Fungi)
cellular_component unknown
extracellular
intracellular
membrane
protein serine/threonine phos...
stroma
site of polarized growth
GO Biological Process (level 6)
4-nitrophenylphosphatase
ABC-type uptake permease

Annotation to All Nodes Layout Add Edges

Dismiss

Figure 17 Cytoscape. (a) An interaction network with a menu displaying available layout algorithms. (b) Mapping control associating node color with expression values for an experimental condition. (c) Detailed attributes for selected nodes and edges. (d) Control for selecting a level in a hierarchy to be used for node and edge annotations. (Reprinted with permission from Ref. [160], © Cold Spring Harbor Laboratory Press.)

a fisheye display, which preferentially allocates space to a selected pathway element and its immediate neighbors, pushing other nodes to the periphery.

The challenges of managing and interpreting increasingly complex interaction diagrams have led some researchers to develop formal languages for pathway languages. PATIKA [50] defines a formal ontology for pathways, with each state and transition as a separate node in the graph. This ontology includes support of nested pathways and provisions for representation of incomplete information. CellDesigner is a network editing tool based on a formal process diagram language that uses visual notations to differentiate between specific types of states, molecules, and reactions [104]. CellDesigner can export model descriptions written in the Systems Biology Markup Language [91] and integrate with the Systems Biology Workbench [90] for simulation of models.

Integration of gene expression data can increase the utility of pathway visualizations. GenMAPP [49] is a tool for creating pathways with nodes that can be color-coded with gene expression values. An associated web site (http://www.genmapp.org) provides a public pathway repository. The companion program MAPPFinder [53] can be used to incorporate gene ontology annotations with GenMAPP pathways visualization. VitaPad [85] supports the creation and visualization of pathways that include edge decorations for catalyzing enzymes or genes and microarray data, along with a relational model for storage of pathway information. DRAGON View [24] is a web-based tool that provides several displays of microarray data, including the use of expression data to color-code nodes in KEGG pathways. ArrayXPath [40] uses diverse databases to produce a generalizable scheme for pathway visualization of microarray data. Given a microarray data set, ArrayXPath uses public databases to resolve probe identifiers, finds appropriate pathways, maps between the microarray data and the pathways, and produces a navigable annotated pathway.

GeneVis [10] uses a 3-D approach to display the hierarchical nature of some regulatory networks. Individual levels in a hierarchical network are drawn as rings, with links between rings indicating regulatory relationships. Distortion can be used to allocate more screen space to levels of interest within the hierarchy.

Several commercial products provide pathway and network visualizations. PimRider and PimWalker (http://pim.hybrigenics.com) are interactive tools for the exploration of protein–protein interaction networks. Pathway Enterprise from OmniViz (http://www.omniviz.com) is a pathway visualization and editing tool that supports a wide variety of pathway content. Ariadne Genomics (http://www.ariadnegenomics.com) provides a suite of tools – PathwayExpert, PathwayStudio Central and PathwayAssist – to support the visualization of pathways, including integration of experimental results and literature analyses.

A comparative evaluation of pathway visualization systems identified several requirements that would increase the utility of these tools for biologists. Automated pathway construction and maintenance based on analysis of scientific literature, linked overview and analysis of multiple pathways, and support for higher-level of abstraction were seen as desirable features [151]. See Table 3.

Table 3 Selected visualizations of interaction networks and pathways

System type	Examples
Graph-based network visualization	[34], PIMViewer [46] PIMWalker [63], VisANT [87], Osprey [25], ProViz [93], Cytoscape [160], GSCope [185], PATIKA [50], GeneVis [10], Genomic Object Net [131], CellDesigner [131] PimRider and PimWalker (http://pim.hybrigenics.com) Pathway Enterprise (http://www.omniviz.com) Pathway tools (http://www.ariadnegenomics.com)
Network and gene expression data	GenMAPP [49], MAPPFinder [53], VitaPad [85], DRAGON View [24], ArrayXPath [40]

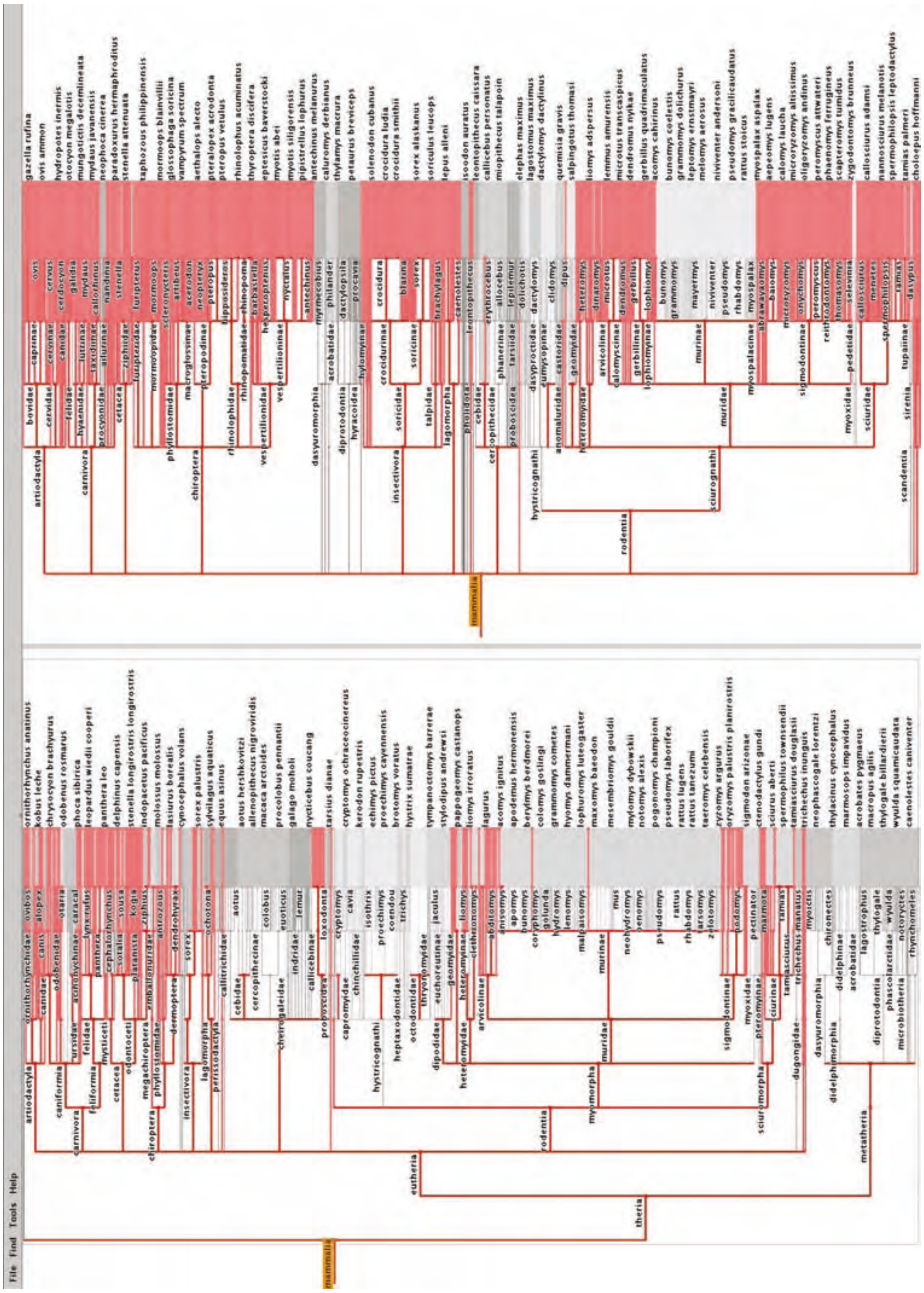
3.5 Phylogenies and Taxonomies

The examination of trees resulting from taxonomic classification and phylogenetic reconstruction is a long-standing challenge. Like protein–protein interaction networks, these trees are often quite large, involving thousands of nodes. Common user tasks generally involve browsing, searching for a known item and examining it in context, filtering to specific subtrees, and comparison of multiple trees [71, 129, 135, 145]. Although most of these tools are based on variants of traditional 2-D tree layouts, alternatives such as hyperbolic displays [92] and immersive virtual reality environments [147] have also been proposed.

TaxonTree [135] combines integrated navigation and visualization of these trees with smoothly animated transitions that illustrate how the tree changes when subtrees are expanded and collapsed. TaxonTree also provides visual context in the form of a highlighted path from the selected node back to the root. Edges can optionally be annotated with markers for shared characteristics used to diagnose phylogenetic relationships. A companion program, DoubleTree, extends TaxonTree to support tree comparison in two adjacent panes with linked coordination.

TreeJuxtaposer [129] takes a different approach to comparisons between these large trees, through the use of a similarity measure between nodes that allows for identification of a node that is most similar to another node. Constant-time lookup of these nodes during interaction provides the basis for linked highlighting of subtrees of other trees that are similar to a highlighted tree (Figure 18). TreeJuxtaposer also provides a distorted display that emphasizes marked areas, guaranteeing that they are always displayed. Progressive rendering features provide rendering times of less than 2 s for trees of 500 000 nodes. TreeJuxtaposer can be used to compare up to four trees at once.

Graham and Kennedy [71] propose an alternative approach to comparative visualization of multiple taxonomies. Taking advantage of the *n*-ary nature of taxonomies (as opposed to generally binary phylogenetic trees), their system


File Find Tools Help
mammalia
theria
eutheria
metatheria
artiodactyla
carnivora
caniformia
feliformia
cetacea
odontoceti
mysticeti
chiroptera
megachiroptera
insectivora
lagomorpha
perissodactyla
proboscidea
rodentia
hystricognathi
sciurognathi
myomorpha
sciuromorpha
dasyuromorphia
didelphimorphia
diprotodontia
microbiotheria
dugongidae
sirenia
scandentia
primates
ornithorhynchus anatinus
kobus leche
chrysocyon brachyurus
odobenus rosmarus
phoca sibirica
leopardus wiedii cooperi
panthera leo
delphinus capensis
stenella longirostris longirostris
indopacetus pacificus
molossus molossus
lasiurus borealis
cynocephalus volans
sorex palustris
sylvilagus aquaticus
equus asinus
aotus hershkovitzi
allenopithecus nigroviridis
macaca arctoides
procolobus pennantii
galago moholi
nycticebus coucang
tarsius dianae
cryptomys ochraceocinereus
kerodon rupestris
echimys pictus
proechimys cayennensis
brotomys voratus
hystrix sumatrae
tympanoctomys barrerae
stylodipus andrewsi
pappogeomys castanops
liomys irroratus
lagurus
acomys ignitus
apodemus hermonensis
berylmys berdmorei
colomys goslingi
grammomys cometes
hyomys dammermani
lophuromys luteogaster
maxomys baeodon
mesembriomys gouldii
mylomys dybowskii
notomys alexis
pogonomys championi
pseudomys laborifex
rattus lugens
rattus tanezumi
taeromys celebensis
zyzomys argurus
oryzomys palustris planirostris
sigmodon arizonae
ctenodactylus gundi
sciurus aberti
spermophilus townsendii
tamiasciurus douglasii
trichechus inunguis
neophascogale lorentzi
thylacinus cynocephalus
marmosops impavidus
acrobates pygmaeus
macropus agilis
thylogale billardierii
wyulda squamicaudata
caenolestes caniventer
gazella rufina
ovis ammon
hydropotes inermis
otocyon megalotis
mungotictis decemlineata
mydaus javanensis
neophoca cinerea
paradoxurus hermaphroditus
stenella attenuata
taphozous philippinensis
mormoops blainvillii
glossophaga soricina
vampyrum spectrum
aethalops alecto
pteralopex acrodonta
pteropus vetulus
rhinolophus acuminatus
thyroptera discifera
eptesicus bavarstocki
myotis abei
myotis siligorensis
pipistrellus lophurus
antechinus melanurus
caluromys derbianus
thylamys macrura
petaurus breviceps
solenodon cubanus
crocidura ludia
crocidura smithii
sorex alaskanus
soriculus leucops
lepus alleni
isoodon auratus
leontopithecus caissara
callicebus personatus
miopithecus talapoin
elephas maximus
lagostomus maximus
dactylomys dactylinus
quemisia gravis
salpingotus thomasi
liomys adspersus
lemmus amurensis
microtus transcaspicus
dendromus nyikae
gerbillus quadrimaculatus
acomys cahirinus
bunomys coelestis
grammomys dolichurus
leptomys ernstmayri
melomys aerosus
niviventer andersoni
pseudomys gracilicaudatus
rattus stoicus
myospalax aspalax
aepeomys lugens
calomys laucha
microryzomys altissimus
oligoryzomys andinus
peromyscus attwateri
phaenomys ferrugineus
scapteromys tumidus
zygodontomys brunneus
callosciurus adamsi
nannosciurus melanotis
spermophilopsis leptodactylus
tamias palmeri
choloepus hoffmanni

uses horizontal labels to display internal nodes in taxonomies. Each label spans the width of its enclosed subtrees, with leaf nodes displayed in grids. Several taxonomies can be stacked vertically, with linked selection and mouse-over brushing showing the position of nodes from a selected subtree in other subtrees. Internal nodes are also annotated with bar charts indicating the portion of a subtree that is currently selected and textures indicating changes in a subtree.

3.6 Phenotypes and Lineages

Phenotypic and cell relationship information provide an ideal opportunity for combining information visualization and microscopic visualization to provide a unified visualization. The RNAi database [73] is a searchable, web-based data repository which displays detailed information on RNAi experiments, including clickable maps of genetic regions and movies of appropriate phenotypes. The associated PhenoBlast tool compares RNAi experiments based on similarity and provides a color-coded result grid that is similar to a heatmap.

The Open Microscopy Environment (OME, discussed below) [70] is a database-driven system for imaging informatics, including tools for manually or automatically classifying images by phenotype. Images can be viewed in graphical browsers that use phenotypes to group images. BioSig [136] provides similar functionality for displaying phenotypic characterizations, using data-driven web interfaces, a navigable data model display, and a query manager that supports "query by feature", in which average values of computed features are used as query values.

Other tools focus more specifically on visualizing the structure of phenotypic or development graphs. PGViewer [178] and CRAVE [69] provide tree-based interfaces for accessing phenotype data from user-defined queries on a data store and ontology, respectively. The Edinburgh Mouse Atlas Projects' prototype visualization of mouse embryo development stages [48] displays developmental stages as acyclic directed graphs, with search filters, region-of-interest selection and zooming facilities. A 3-D version of the tool displays multiple data sets in parallel planes, with links between the planes indicating relationships between data sets.

Figure 18 TreeJuxtaposer. Subtrees in the left-hand tree are marked in corresponding colors in the right-hand tree. Guaranteed visibility ensures that these marked areas are visible in both trees. (From Ref. [129], used by permission.)

3.7 Visualization of the Scientific Process

Most of the visualizations described in this chapter focus on experimental results. Tools for visualizing both the processes involved in scientific analyses, and the conclusions that can be drawn from those analyses, can help bioinformaticians bridge the gap between analysis and synthesis.

Data analysis often involves the repeated application of a series of processes to new instances of a type of data. For example, genomics analyses might involve searching for data in one or more databases, followed by one or more transformation of results or subsequent searches [170]. Like raw experimental data, these workflows are valuable artifacts of the scientific process. Graphical tools for creating these workflows provide users with the ability to place computational modules on a canvas and to connect them with links that specify data flow. Examples include the Open Microscopy Environment's "Chain Builder" [70] (Figure 19) and the Taverna [133] tool for genomic analysis workflows.

Once collected and analyzed, data must be interpreted to evaluate hypotheses and develop new models. "Scientific notebook" tools attempt to augment data storage and management with interpretive data and model construction. Biological Storytelling [107] integrates a data display with a story editor that can be used to collect evidence for and against various interpretations, and a diagram editor for management of a network representation of the data interpretation.

Other visualizations examine use the content of research publications to inform further investigation. Cocitation analyses and related strategies have been used to generate models and visualizations of relationships between terms and concepts in published research in bioinformatics (and many other fields). Examples include networks of gene relationships based on co-occurrences in Medline records and color coded based on gene expression values [95], and networks that analyze publications trends over many years [121].

4 Image Informatics

The microscopy image techniques described at the start of this chapter are primarily concerned with data collection, whereas information visualization tools support interpretation and analysis of data. Image informatics bridges the gap between microscopy data collection and image interpretation by providing a framework for associating images with structured meta-data describing acquisition parameters. This information provides context that is necessary for both manual and computational interpretation of the images.

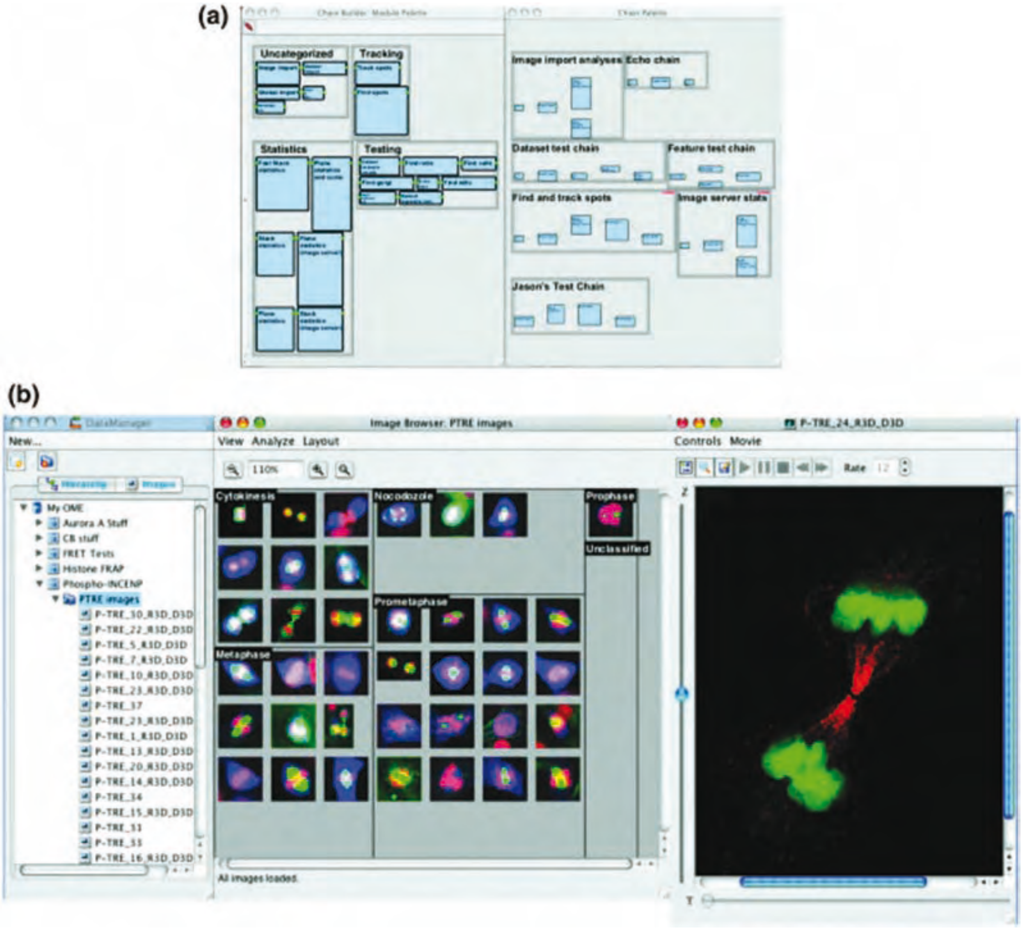


Figure 19 Visualization of microscopy data and analytic components in the OME. (a) The Chain Builder: computational modules (left) can be combined into analysis chains or workflows (right). (b) A data manager for hierarchical navigation of projects and images sets, a view of an individual image set, laid out according to manual categorization and an image viewer. (From Ref. [70], used by permission.)

This requires tools such as VisBio for displaying the image content, analytic tools for computational analysis of both images and meta-data, and information visualization tools for examining meta-data associated with the images.

An image without meta-data is of little scientific value because it cannot be meaningfully interpreted. Even a small collection of images requires attached meta-data in order to provide context and allow the image contents to be informative. What is this an image of? How was it acquired? By whom? What was the experiment? With large collections of images from high-throughput or genomic studies, the scientific conclusions that an image represents must also be recorded and associated with the relevant images due to the sheer number of them involved. If a conclusion was arrived at by image analysis, the steps taken to perform the analysis including any parameters and intermediate results must similarly be associated with the images analyzed. If the conclusion was arrived at by manual inspection, then the identity of the person making the conclusion must be recorded, including the option of maintaining multiple, possibly contradictory conclusions by different individuals. The management of this information about images, which is necessary for their scientific interpretation, is referred to as Image Informatics.

The need to store, analyze, retrieve and interpret this information presents significant challenges. The resulting systems have complex data models that may in themselves require visualization – in addition to visualizing data, users may need visual tools to inspect and explore data models, in order to find data of interest and draw comparison between different data sets. An examination of the specific issues associated with image informatics tools illustrates some of the challenges presented by these systems

An image informatics infrastructure for HCS should use an extensible data model to manage images and image meta-data. Extraction of scientifically meaningful results from large image collections will require an image analysis framework that allows for easy customization by the end-user biologist. The infrastructure must maintain an audit trail of any results it maintains such that everything can be traced back to an acquisition system or another entry point into the system. Open access to raw data, meta-data and analysis results will be needed to incorporate data from different acquisition systems and technologies, to compare results from differing analytic techniques, and to integrate image data with information in relevant external data sources. User tools – including both microscopy and information visualization – will be needed to allow scientists to easily manage, retrieve, manipulate, annotate, and interpret this data.

In traditional software for microscopy many of these requirements are met on an ad hoc basis. However, the high-throughput nature of HCS precludes ad hoc solutions and demands that these issues be treated systematically. The only known system to integrate all of these requirements in one place is the OME (http://openmicroscopy.org [70, 175]). See Chapter 44 for additional perspectives on workflow, provenance and other challenges addressed in this section.

4.1 Data and Information Management

All information for HCS must be stored in a relational database management system (RDBMS). The structure of the RDMS (the DB schema) must be specifically designed for HCS, which poses several challenges in DB design.

The first challenge is that image data comes in two highly distinct forms. The pixel data is very large and relatively unstructured while the meta-data and any extracted information is relatively smaller, but highly structured. These two forms are sufficiently different that storing them in common is impractical. Specifically, RBDMS binary data storage and retrieval facilities do not provide the performance and accessibility needed to rapidly retrieve arbitrary slices or time points from 5-D microscopy data. This shortcoming can be addressed by storing meta-data in the RDBMS, along with appropriate pointers to an image store on a private file system not accessible to normal users (a pixel repository). A specialized image server can support flexible retrieval of image data from the repository.

The second challenge is much harder to overcome and it is related to a general problem of databases used in scientific research – it is not possible to completely define the database structure when the database is being designed. The database is used in the discovery process, which leads to new information that must be stored in the same database. Although relational databases are extensible, this flexibility is at too low a level to be useful to most users. Augmenting the database with metadata that describes the structure of the raw scientific data provides application software with information needed to determine how to interpret and present that data. Interactive tools for creating new data types can help biologists extend the database structure, without having to learn the details of RDBMS query and data definition languages.

4.2 Image Analysis

Although image analysis has been around decades, the advent of HCS places new demands on this traditional field. Image analysis for microscopy is extremely difficult because of the bewildering variety of morphologies that biology can produce. As attempts to provide all algorithms to address all possible morphologies are unrealistic, a more flexible approach is necessary. Specifically, basic tools can be combined with mechanisms for connecting basic analyses and integrating external, possibly user-developed, possibly legacy tools.

There are two fundamentally different approaches to image analysis: object identification and scene-based analysis. Almost all image analysis software for microscopy is based on object identification. In this approach, the object of interest is identified and separated (segmented) away from its background.

The object of interest can be a nucleus or a cell body, a subcellular organelle, etc. This approach is very natural for fluorescence microscopy because a specific fluorescent probe generates the contrast in the image, and it is natural to extract everything possible about this fluorescence signal: its distribution, intensity, shape, etc. In practice, however, the picture is never as clear as it seems it should be. Segmentation algorithms to identify objects of interest are easily fooled by irrelevant objects, fluctuations in pixel intensity that must somehow be “normalized”, the object’s shape and other characteristics that cannot be easily controlled experimentally. Algorithms that can reliably identify individual cells regardless of cell type, the manner in which they grow and the specific probe used to visualize them are extremely difficult to develop. The advantage they have over scene-based analysis is that when they do work, they can provide extremely valuable quantitative information about the objects they identify [32, 140, 141].

Scene analysis is new to microscopy, but can be thought of as a natural as well. Consider an image of a field of cells. Is it an image of one cell repeated many times, or can we analyze the field of cells as a whole? Since this approach does not rely on identifying specific objects, it can be much more robust, i.e. applicable to a much wider array of phenotypes displayed in HCS (Figure 20). Scene-based analysis relies on pattern recognition techniques or machine learning to train an image classifier. In supervised machine learning, the classifier is trained to distinguish classes of images defined as a set of negative controls and one or more sets of target phenotypes (for example, “Ruffled” and “Not Ruffled” in Figure 20). Once trained using several example images of each class, the classifier can then be used in an experiment to score images as being similar to the negative control or one of the target phenotypes. There are presently no well-characterized examples of unsupervised learning for biological image analysis, but if a reliable and relevant measure of image similarity can be determined, standard clustering algorithms can be used to automatically define phenotypic classes without manual intervention [18–20].

Image analysis systems for biology must provide entry points for extension. This requires that the data model upon which they operate must be open and well documented. It must be possible for the end-users to interject their own code at essentially any point in the analysis process. Closed monolithic systems preclude this. If there are ways for the end-user to interject their own code, then much of the system can still be used even if it was not designed to solve the particular problem at hand.

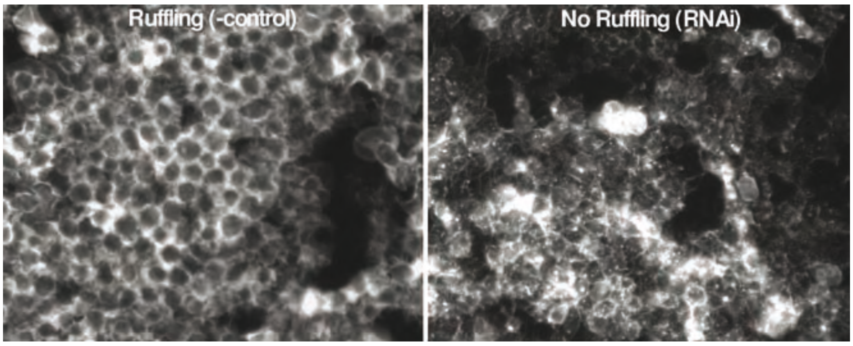


Figure 20 The ruffled appearance of these cells is caused by the extension of filapodia, which are used for locomotion. The filapodia are visualized using a fluorescent probe for the actin protein, which is a major component of these structures. The image on the right is the result of disrupting a gene required for the formation of the morphologically normal filapodia displayed on the left. These two morphologies are easily distinguished using a Bayes net classifier analyzing the field of view as a whole (10/10 images classified correctly with 85% confidence), while posing a significant challenge to object identification methods. (Images from Pamela Bradley, Harvard Medical School; image analysis and classification by the Goldberg lab at NIH.)

4.3 Analysis Workflows

Analysis protocols, in which independent computational algorithms are chained together to derive desired results, have been effectively used in a variety of fields, including genomic analysis [133, 170]. An image informatics platform can store the definitions of these protocols along with the results they produce, thus supporting reuse and comparison of results. Appropriate user tools can be used to modify and extend existing workflows for new screens. End-user tools for browsing, searching libraries of individual components and workflows can give biologists complete control over the image analysis, and a great deal of flexibility without necessarily involving additional software developers.

4.4 Provenance

The challenge of data provenance (or *lineage* – the storage and retrieval of the origins and computational history of data and results) has been extensively studied [21].

HCS informatics systems contain many possible kinds of analyses that can generate similar results. The results differ in their context even if they carry the same semantic meaning. If two different segmentation algorithms generate different particle counts, then neither count can be considered invalid or meaningless – these two conflicting scores differ somehow in the data dependency chain that led to the conflicting results. The ability to reconstruct these dependency chains and determine the provenance of a given piece of information is an essential component of an HCS informatics system. The visualization of these chains is an ongoing research effort, and it is expected that direct visualization of information flow through an HCS system would be an invaluable tool for the scientist to pick out new patterns of information and make most effective use of existing analysis tools.

4.5 Federation

Federation of information with other information sources is necessary for linking image information to data that are beyond the scope of the HCS system. It is not possible to make hard and fast rules where this transition should take place. The HCS system must be capable of maintaining references to external data sources and providing them with sufficient context so that experimental results can be easily viewed within the context of another informatics system. For example, in a compound screen, it must be possible to take a collection of hits determined by the HCS and send the list of compounds they resulted from to a cheminformatics system in order to determine the structure–activity relationships for these compounds. Similarly, for an RNAi or other genomic screen, it must be possible to ascertain what is known about the functions of the genes that were scored as hits. Clearly these are not within the scope of an HCS system, so it must maintain references to compounds and genes in order to perform these tasks using other information sources. For general issues in data federation, see also Chapter 41.

4.6 Visualization and User Tools

Effective use of an informatics system for HCS requires tools that biologists can use to locate, manage, manipulate, and interpret various types of data. Given thousands of images, divided into hundreds of sets for various experiments, simply locating raw data might be challenging, particularly when the data is not recent. Similarly, the presence of numerous analytic modules, workflows, and data types for analytic results poses challenges in navigation and searching. An integrated, extensible environment that provides linked, coordinated tools for these tasks will provide multiple perspectives on the various types, along with tools for manipulating them as necessary (Figure 20).

5 Conclusion: Research Questions and Challenges

Effective visualizations leverage the power of the human visual and perceptual system to amplify cognition [31]. Visualizations of biological data can help scientists interpret experimental results, evaluate hypotheses, generate new hypotheses and build models. Maximizing the potential of these visualizations will require development of tools that are appropriately designed to support meaningful tasks, flexible in their management and presentation of data, and integrated with other visualizations, data sources and tools.

Appropriately designed tools will be based on detailed understanding of user needs for specific data sets and tasks. Although most of the systems described in this chapter were designed to meet specific needs, descriptions of explicit needs assessments are relatively rare. Task-specific analyses of user needs can provide invaluable guidance for visualization design. Questionnaires [170] and interviews [151] are standard techniques that have proven useful for this purpose.

Evaluation is a crucial element of appropriate visualization design. Given multiple different approaches to visualization of a given data type, which (if any) is best? Two-dimensional [135], hyperbolic [92] and 3-D immersive [147] approaches have been proposed for phylogenetic and taxonomic trees. Given multiple different approaches to visualization of a given data type, which (if any) best? Without some means of identifying strengths and weaknesses of various techniques, researchers will be limited in their ability to constructively build upon prior work, or to select the most appropriate tools for their own analysis. Although controlled user studies [150] and heuristic evaluation by expert users [48, 151] have been applied to some tools, evaluation is relatively rare. Related work in human–computer interaction and information visualization research may prove useful to developers of biological visualizations: the relative strengths of 2- versus 3-D interfaces have been repeatedly examined [43] and the development of evaluation techniques for visualizations has been identified as a general challenge in information visualization [139].

Many of the visualization systems described above are limited in their flexibility. Although some systems provide linked alternative views of a data type, others are limited to a single visual representation. Many proposals tightly couple a visualization tool to a novel analysis algorithm.

Tools that support the analysis of data via multiple different linked representations may help users find patterns that would not have been visible in any single view. Support for ad hoc coordination of alternative views might be particularly powerful in this regard [132]. Decoupling of analysis algorithms from visualization tools can help scientists make meaningful comparisons between competing analytic approaches.

Extensible visualization environments and toolkits of reusable components can facilitate development of these systems. Third-party developers have used plugin architectures for Cytoscape [160] and ImageJ (http://rsb.info.nih.gov/ij) to substantially augment the capabilities of the base tools. General-purpose information visualization toolkits [12, 59, 77], as well as special-purpose toolkits for bioinformatics visualization such as GenoViz (http://genoviz.sourceforge.net) and the genome data visualization toolkit [174], can simplify system development substantially.

A final form of flexibility involves innovative use of display technologies. Tools that take advantage of high-resolution desktop displays [129], wall-sized displays [82] and immersive environments [147] provide opportunities for display of larger data sets and increased collaboration between users.

Integrated visualizations of multiple data types can provide interpretive power not available in simpler tools. Examples including multi-track genome browsers [89, 97] and microarray visualizations augmented with GO annotations [8] illustrate the possibilities, but much more can be done in this regard. Approaches such as the "omic space" model [186], which integrates several types of data in one display, may be quite powerful. Another useful approach to integration might involve connections between visualizations and accepted models for experimental and related data, perhaps including microarray [166] and microscopy [70] data models.

Visualizations should also be integrated with the experimental process as a whole. Combining with systems biology tools [90], analysis workflow tools [70,133], and analysis and data storage platforms [181] raises a number of challenges similar to those described in the above discussion of image informatics. However, the resulting systems will increase the power of visualization by providing scientists with the ability to examine raw and interpreted data and metadata in a coordinated manner. Such combinations will provide rich context and interactivity necessary for active exploration and generation of scientific insight.

References

1 Abramoff, M. D., P. J. Magalhaes and S. J. Sam. 2004. Image processing with ImageJ. Biophotonics Int. **11(7)**: 36–42.

2 Ahlberg, C. and B. Shneiderman. 1994. Visual information seeking: tight coupling of dynamic query filters with starfield displays. In Proc. ACM CHI Conference on Human Factors in Computing Systems, Boston, MA, USA: 313–7.

3 Alfarano, C., C. E. Andrade, K. Anthony, et al. 2005. The Biomolecular Interaction Network Database and related tools 2005 update. Nucleic Acids Res. **33**: D418–24.

4 Ashburner, M., C. A. Ball, J. A. Blake, et al. 2000. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet. **25**: 25–9.

5 Awad, I. A., C. A. Rees, T. Hernandez-Boussard, C. A. Ball and G. Sherlock. 2004. Caryoscope: an open source Java application for viewing micoarray data in a genomic context. BMC Bioinformatics **5**: 151.

6 Azuaje, F., H. Wang and A. Chesneau. 2005. Non-linear mapping for exploratory data analysis in functional genomics. BMC Bioinformatics **6**: 13.

7 Bader, J. S., A. Chaudhuri, J. M. Rothberg and J. Chant. 2004. Gaining

confidence in high-throughput protein interaction networks. Nat. Biotechnol. **22**: 78–85.

8 Baehrecke, E. H., N. Dang, K. Barbaria and B. Shneiderman. 2004. Visualization and analysis of microarray and gene ontology data with treemaps. BMC Bioinformatics **5**: 84.

9 Baerends, R. J. S., W. K. Smits, A. d. Jong, L. W. Hamoen, J. Klok and O. P. Kuipers. 2004. Genome2D: a visualization tool for the rapid analysis of bacterial transcriptome data. Genome Biol. **5**: R37.

10 Baker, C. A. H., M. S. T. Carpendale, P. Prusinkiewicz and M. G. Surette. 2002. GeneVis: visualization tools for genetic regulatory network dynamics. Presented at IEEE Visualization '02, Boston, MA, USA: 243–50.

11 Bastiaens, P. I. H. and A. Squire. 1999. Fluorescence lifetime imaging microscopy: spatial resolution of biochemical processes in the cell. Trends Cell Biol. **9**: 48–52.

12 Bederson, B. B., J. Grosjean and J. Meyer. 2004. Toolkit design for interactive structured graphics. IEEE Trans. Software Eng. **30**: 535–46.

13 Bederson, B. B. and J. D. Hollan. 1994. Pad++: a zooming graphical interface for exploring alternative interface physics. In Proc. ACM Symp. on User Interface Software and Technology, Santa Fe, NM, USA: 17–26.

14 Bederson, B. B. and B. Shneiderman (eds.). 2003. *The Craft of Information Visualization: Readings and Reflections*. Morgan Kauffman, San Francisco, CA.

15 Bird, D. K., K. W. Eliceiri, C. H. Fan and J. G. White. 2004. Simultaneous two-photon spectral and lifetime fluorescence microscopy. Appl. Opt. **43**: 5173–82.

16 Bird, D. K., L. Yan, K. M. Vrotsos, K. W. Eliceiri, E. M. Vaughan, P. J. Keely, J. G. White and N. Ramanujam. 2005. Metabolic mapping of MCF10A human breast cells via multiphoton fluorescence lifetime imaging of the coenzyme NADH. Cancer Res. **65**: 8766–73.

17 Birney, E., T. D. Andrews, P. Bevan, et al. 2004. An overview of Ensembl. Genome Res. **14**: 925–8.

18 Boland, M. V., M. K. Markey and R. F. Murphy. 1998. Automated recognition of patterns characteristic of subcellular structures in fluorescence microscopy images. Cytometry **33**: 366–75.

19 Boland, M. V. and R. F. Murphy. 2001. A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. Bioinformatics **17**: 1213–23.

20 Boland, M. V. and R. F. Murphy. 1999. Automated analysis of patterns in fluorescence-microscope images. Trends Cell. Biol. **9**: 201–2.

21 Bose, R. and J. Frew. 2005. Lineage retrieval for scientific data processing. ACM Comput. Surv. **37**: 1–28.

22 Both, M., M. Vogel, O. Friedrich, F. von Wegner, T. Kunsting, R. H. Fink and D. Uttenweiler. 2004. Second harmonic imaging of intrinsic signals in muscle fibers *in situ*. J. Biomed. Opt. **9**: 882–92.

23 Boulesteix, T., E. Beaurepaire, M. P. Sauviat and M. C. Schanne-Klein. 2004. Second-harmonic microscopy of unstained living cardiac myocytes: measurements of sarcomere length with 20-nm accuracy. Opt. Lett. **29**: 2031–3.

24 Bouton, C. M. L. S. and J. Pevsner. 2002. DRAGON View: Information visualization for annotated microarray data. Bioinformatics **18**: 323–4.

25 Breitkreutz, B.-J., C. Stark and M. Tyers. 2003. Osprey: a network visualization system. Genome Biol. **4**: R22.

26 Brown, E., T. McKee, E. diTomaso, A. Pluen, B. Seed, Y. Boucher and R. K. Jain. 2003. Dynamic imaging of collagen and its modulation in tumors *in vivo* using second-harmonic generation. Nat. Med. **9**: 796–800.

27 Brummelkamp, T. R., R. Bernards and R. Agami. 2002. A system for stable expression of short interfering RNAs in mammalian cells. Science **296**: 550–3.

28 Bu, D., Y. Zhao, L. Cai, et al. 2003. Topological structure analysis of the protein-protein interaction network in budding yeast. Nucleic Acids Res. **31**: 2443–50.

29 CAMPAGNOLA, P. J. AND L. M. LOEW. 2003. Second-harmonic imaging microscopy for visualizing biomolecular arrays in cells, tissues and organisms. Nat. Biotechnol. **21**: 1356–60.

30 CANNON, S. B., A. KOZIK, B. CHAN, R. MICHELMORE AND N. D. YOUNG. 2003. DiagHunter and GenoPix2D: programs for genomic comparisons, large-scale homology discovery and visualization. Genome Biol. **4**: R68.

31 CARD, S., J. MACKINLAY AND B. SHNEIDERMAN. 1999. *Using Vision to Think: Readings in Information Visualization*. Morgan Kaufmann, San Francisco, CA

32 CARPENTER, A. E., T. R. JONES AND D. M. SABATINI. 2004. CellProfiler: Cell Image Analysis Software. [Online] http://www.cellprofiler.org (Accessed Sep. 20, 2006).

33 CENTONZE, V. E. AND J. G. WHITE. 1998. Multiphoton excitation provides optical sections from deeper within scattering specimens than confocal imaging. Biophys. J. **75**: 2015–24.

34 CHANG, A. N., J. MCDERMOTT AND R. SAMUDRALA. 2005. An enhanced java graph applet interface for visualizing interactomes. Bioinformatics **21**: 1741–42.

35 CHEN, F.-C. AND T.-J. CHUANG. 2005. ESTviewer: a web interface for visualizing mouse, rat, cattle, pig and chicken conserved ESTs in human genes and human alternatively spliced variants. Bioinformatics **21**: 2510–3.

36 CHEN, H., D. D. HUGHES, T. A. CHAN, J. W. SEDAT AND D. A. AGARD. 1996. IVE (Image Visualization Environment): a software platform for all three-dimensional microscopy applications. J. Struct. Biol. **116**: 56–60.

37 CHEN, W., F. ERDOGAN, H.-H. ROPERS, S. LENZNER AND R. ULLMANN. 2005. CGHPRO – A comprehensive data analysis tool for array CGH. BMC Bioinformatics **6**: 85.

38 CHI, B., R. J. DELEEUW, B. P. COE, C. MACAULAY AND W. L. LAM. 2004. SeeGH – a software tool for visualization of whole genome array comparative genomic hybridization data. BMC Bioinformatics **5**: 13.

39 CHI, E. H.-H., J. RIEDL, E. SHOOP, J. V. CARLIS, E. RETZEL AND P. BARRY. 1996. Flexible Information Visualization of Multivariate Data from Biological Sequence Similarity Searches. Presented at the 7th IEEE Conf. on Visualization '96, San Francisco, CA, USA: 133–40.

40 CHUNG, H.-J., M. KIM, C. H. PARK, J. KIM AND J. H. KIM. 2004. ArrayXPath: mapping and visualizing microarray gene-expression data with integrated biological pathway resources using Scalar Vector Graphics. Nucleic Acids Res. **32**: W460–4.

41 CLAMP, M., D. ANDREWS, D. BARKER, et al. 2003. Ensembl 2002: accommodating comparative genomics. Nucleic Acids Res. **31**: 38–42.

42 CLENDENON, J. L., C. L. PHILLIPS, R. M. SANDOVAL, S. FANG AND K. W. DUNN. 2002. Voxx: a PC-based, near real-time volume rendering system for biological microscopy. Am. J. Physiol. Cell Physiol. **282**: C213–8.

43 COCKBURN, A. AND B. MCKENZIE. 2002. Evaluating the effectiveness of spatial memory in 2D and 3D physical and virtual environments. Presented at the CHI Conf. on Human Factors in Computing Systems, Minneapolis, MN, USA: 203–10.

44 COGHLAN, L., U. UTZINGER, R. RICHARDS-KORTUM, C. BROOKNER, A. ZULUAGA, I. GIMENEZ-CONTI AND M. FOLLEN. 2001. Fluorescence spectroscopy of epithelial tissue throughout the dysplasia-carcinoma sequence in an animal model: spectroscopic changes precede morphologic changes. Lasers Surg. Med. **29**: 1–10.

45 COLE, M. J., J. SIEGEL, S. E. WEBB, et al. 2001. Time-domain whole-field fluorescence lifetime imaging with optical sectioning. J. Microsc. **203**: 246–57.

46 COLLAND, F., X. JACQ, V. TROUPLIN, et al. 2004. Functional proteomics mapping of a human signaling pathway. Genome Res. **14**: 1324–32.

47 Craig, P., J. Kennedy and A. Cumming. 2005. Animated interval scatter-plot views for the exploratory analysis of large-scale microarray time-course data. Inf. Visual. **4(3)**: 149–63.

48 Dadzie, A.-S. and A. Burger. 2005. Providing visualisation support for the analysis of anatomy ontology data. BMC Bioinformatics **6**: 74.

49 Dahlquist, K. D., N. Salomonis, K. Vranizan, S. C. Lawlor and B. R. Conklin. 2002. GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. Nat. Genet. **31(1)**: 19–20.

50 Demir, E., O. Babur, U. Dogrusoz, A. Gursoy, G. Nisanci, R. Cetin-Atalay and M. Ozturk. 2002. PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways. Bioinformatics **18**: 996–1003.

51 Denk, W., J. H. Strickler and W. W. Webb. 1990. Two-photon laser scanning fluorescence microscopy. Science **248**: 73–6.

52 Dickinson, M. E., E. Simbuerger, B. Zimmermann, C. W. Waters and S. E. Fraser. 2003. Multiphoton excitation spectra in biological samples. J. Biomed. Opt. **8**: 329–38.

53 Doniger, S. W., N. Salomonis, K. D. Dahlquist, K. Vranizan, S. C. Lawlor and B. R. Conklin. 2003. MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. Genome Biol. **4(1)**: R7.

54 Eichler, G. S., S. Hang and D. Ingber. 2003. Gene Expression Dynamics Inspector (GEDI): for intergrative analysis of expression profiles. Bioinformatics **19**: 2321–22.

55 Eisen, M. B., P. T. Spellman, P. O. Brown and D. Botstein. 1998. Cluster analysis and display of genome-wide expression patterns. Proc. Natl Acad. Sci. USA **95**: 14863–8.

56 Ekdahl, S. and E. L. L. Sonnhammer. 2004. ChromoWheel: a new spin on eukaryotic chromosome visualization. Bioinformatics **20**: 576–7.

57 Elbashir, S. M., J. Harborth, W. Lendeckel, A. Yalcin, K. Weber and T. Tuschl. 2001. Duplexes of 21-nucleotide RNAs mediate RNA interference in cultured mammalian cells. Nature **411**: 494–8.

58 Fang, Z., K. Cone, H. Sanchez-Villeda, et al. 2003. iMap: a database-driven utility to integrate and access the genetic and physical maps of maize. Bioinformatics **19**: 2105–11.

59 Fekete, J.-D. 2004. The Info Vis Toolkit. Presented at the IEEE Symp. on Information Visualization, Austin, TX, USA: 167–74.

60 Fire, A. 1994. A four-dimensional digital image archiving system for cell lineage tracing and retrospective embryology. Comput. Appl. Biosci. **10**: 443–7.

61 Fire, A., S. Xu, M. K. Montgomery, S. A. Kostas, S. E. Driver and C. C. Mello. 1998. Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*. Nature **391**: 806–11.

62 Fischer, G., S. M. Ibrahim, G. A. Brockmann, J. Pahnke, E. Bartocci, H.-J. Thiesen, P. Serrano-Fernández and S. Möller. 2003. Expressionview: visualization of quantitative trait loci and gene expression data in Ensembl. Genome Biol. **4**: R77.

63 Formstecher, E., S. Aresta, V. Collura, et al. 2005. Protein interaction mapping: a *Drosophila* case study. Genome Res. **15**: 376–84.

64 Frazer, K. A., L. Pachter, A. Poliakov, E. M. Rubin and I. Dubchak. 2004. VISTA: computational tools for comparative genomics. Nucleic Acids Res. **32**: W273–9.

65 Furnas, G. W. 1986. Generalized fisheye views. Presented at the ACM CHI Conf. on Human Factors in Computing Systems, Boston, MA, USA: 16–23.

66 Gadella, T. W. J., T. M. Jovin and R. M. Clegg. 1993. Fluorescence lifetime imaging microscopy (FLIM): spatial resolution of microstructures on the nanosecond time scale. Biophys. Chem. **48**: 231–9.

67 Gerth, V. E. and P. D. Vize. 2005. A Java tool for dynamic web-based 3D visualization of anatomy and overlapping gene or protein expression patterns. Bioinformatics **21**: 1278–9.

68 Giot, L., J. S. Bader, C. Brouwer, et al. 2003. A protein interaction map of *Drosophila melanogaster*. Science **302**: 1727–36.

69 Gkoutos, G. V., E. C. J. Green, S. Greenaway, A. Blake, A.-M. Mallon and J. M. Hancock. 2005. CRAVE: a database, middleware and visualization system for phenotype ontologies. Bioinformatics **21**: 1257–62.

70 Goldberg, I. G., C. Allan, J. M. Burel, et al. 2005. The Open Microscopy Environment (OME) Data Model and XML file: open tools for informatics and quantitative analysis in biological imaging. Genome Biol. **6**: R47.

71 Graham, M. and J. Kennedy. 2005. Extending taxonomic visualization to incorporate synonymy and structural markers. Inf. Visual. **4(3)**: 206–23.

72 Greshock, J., T. Naylor, A. Margolin, et al. 2004. 1-Mb resolution array-based comparative genomic hybridization using a BAC clone set optimized for cancer gene analysis. Genome Res. **14**: 179–87.

73 Gunsalus, K. C., W.-C. Yueh, P. MacMenamin and F. Piano. 2004. RNAiDB and PhenoBlast: web tools for genome-wide phenotypic mapping projects. Nucleic Acids Res. **32**: D406–10.

74 Guo, S. and K. J. Kemphues. 1995. *par-1*, a gene required for establishing polarity in *C. elegans* embryos, encodes a putative Ser/Thr kinase that is asymmetrically distributed. Cell **81**: 611–20.

75 Hammond, S. M., E. Bernstein, D. Beach and G. J. Hannon. 2000. An RNA-directed nuclease mediates post-transcriptional gene silencing in *Drosophila* cells. Nature **404**: 293–6.

76 Havre, S. L., M. Singhai, D. A. Payne and B.-J. M. Webb-Robertson. 2004. PQuad: Visualization of Predicted Peptides and Proteins. Presented at the IEEE Symp. on Information Visualization, Austin, TX, USA: 473–80.

77 Heer, J., S. K. Card and J. A. Landay. 2005. prefuse: a toolkit for interactive information visualization. Presented at the SIGCHI Conf. on Human Factors in Computing Systems, Portland, OR, USA: 421–30.

78 Heilker, R., L. Zemanova, M. J. Valler and G. U. Nienhaus. 2005. Confocal fluorescence microscopy for high-throughput screening of G-protein coupled receptors. Curr. Med. Chem. **12**: 2551–9.

79 Helt, G. A., S. Lewis, A. E. Loraine and G. M. Rubin. 1998. BioViews: Java-based tools for genomic data visualization. Genome Res. **8**: 291–305.

80 Hibbard, W. 1998. VISAD – connecting people to computations and people to people. Comput. Graphics **32**: 10–12.

81 Hibbard, W., S. Emmerson, C. Rueden, T. Rink, D. Glowacki, N. Rasmussen, D. Fulker and J. Anderson. 1999. Collaborative visualization and computation in the earth sciences using VisAD. In Preprints Conf. on Interactive Information and Processing Systems for Meteorology, Oceanography and Hydrology, Dallas, TX: 478–80.

82 Hibbs, M. A., N. C. Dirksen, K. Li and O. G. Troyanskaya. 2005. Visualization methods for statistical analysis of microarray clusters. BMC Bioinformatics **6**: 115.

83 Hochheiser, H., E. H. Baehrecke, S. M. Mount and B. Shneiderman. 2003. Dynamic querying for pattern identification in microarray and genomic data. In Proc. IEEE Multimedia Conf. and Expo, Baltimore, MD, USA, **3**: 453–6.

84 Hochheiser, H. and B. Shneiderman. 2004. Dynamic query tools for time series data sets: timebox widgets for interactive exploration. Inf. Visual. **3**: 1–18.

85 Holford, M., N. Li, P. Nadkarni and H. Zhao. 2005. VitaPad: visualization tools for the analysis of pathway data. Bioinformatics **21**: 1596–602.

86 Hsu, F., T. H. Pringle, R. M. Kuhn, D. Karolchik, M. Diekhans, D. Haussler and W. J. Kent. 2005. The UCSC proteome browser. Nucleic Acids Res. **33**: D454–8.

87 Hu, Z., J. Mello, J. Wu and C. DeLisi. 2004. VisANT: an online visualization and analysis tool for biological interaction data. BMC Bioinformatics **5**: 17.

88 HUANG, Y., J. PUMPHREY AND A. R. GINGLE. 2005. ESTminer: a Web interface for mining EST contig and cluster databases. Bioinformatics **21**: 669–70.

89 HUBBARD, T., D. BARKER, E. BIRNEY, et al. 2002. The Ensembl Genome Database Project. Nucleic Acids Res. **30**: 38–41.

90 HUCKA, M., A. FINNEY, H. M. SAURO, H. BOLOURI, J. DOYLE AND H. KITANO. 2002. The ERATO Systems Biology Workbench: enabling interaction and exchange between software tools for computational biology. Pac. Symp. Biocomput. **7**: 450–61.

91 HUCKA, M., A. FINNEY, H. M. SAURO, et al. 2003. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics **19**: 524–31.

92 HUGHES, T., Y. HYUN AND D. LIBERLES. 2004. Visualising very large phylogenetic trees in three dimensional hyperbolic space. BMC Bioinformatics **5**: 48.

93 IRAGNE, F., M. NIKOLSKI, B. MATHIEU, D. AUBER AND D. SHERMAN. 2003. ProViz: Protein Interaction visualization and exploration. Bioinformatics **21**: 272–4.

94 JAGER, S., L. BRAND AND C. EGGELING. 2003. New fluorescence techniques for high-throughput drug discovery. Curr. Pharm. Biotechnol. **4**: 463–76.

95 JENSSEN, T.-K., A. LÆGREID, J. KOMOROWSKI AND E. HOVIG. 2001. A literature network of human genes for high-throughput analysis of gene expression. Nat. Genet. **28**: 21–8.

96 KANEHISA, M., S. GOTO, S. KAWASHIMA, Y. OKUNO AND M. HATTORI. 2004. The KEGG resource for deciphering the genome. Nucleic Acids Res. **32**: D277–80.

97 KAROLCHIK, D., R. BAERTSCH, M. DIEKHANS, et al. 2003. The UCSC genome browser database. Nucleic Acids Res. **31**: 51–54.

98 KAROLCHIK, D., A. S. HINRICHS, T. S. FUREY, K. M. ROSKIN, C. W. SUGNET, D. HAUSSLER AND W. J. KENT. 2004. The UCSC Table Browser data retrieval tool. Nucleic Acids Res. **32**: D493–6.

99 KARP, P. D. 2001. Pathway databases: a case study in computational symbolic theories. Science **293**: 2040–4.

100 KASHUK, C., S. SENGUPTA, E. EICHLER AND A. CHAKRAVARTI. 2002. viewGene: a graphical tool for polymorphism visualization and characterization. Genome Res. **12**: 333–8.

101 KERKHOVEN, R., F. H. J. VAN ENCKEVORT, J. BOEKHORST, D. MOLENAAR AND R. J. SIEZEN. 2004. Visualization for genomics: the Microbial Genome Viewer. Bioinformatics **20**: 1812–4.

102 KINCAID, R. 2004. VistaClara: an interactive visualization for exploratory analysis of DNA microarrays. Presented at the ACM Symp. on Applied Computing, Nicosia: 167–74.

103 KINCAID, R., A. BEN-DOR AND Z. YAKHINI. 2005. Exploratory visualization of array-based comparative genomic hybridization. Inf. Visual. **4**: 1–15.

104 KITANO, H., A. FUNAHASHI, Y. MATSUOKA AND K. ODA. 2005. Using process diagrams for the graphical representation of biological networks. Nature Biotechnology **23**: 961–6.

105 KOHONEN, T. 1995. *Self-Organizing Maps*. Springer, Berlin.

106 KOZIK, A., E. KOCHETKOVA AND R. MICHELMORE. 2002. GenomePixelizer – a visualization program for comparative genomics within and between species. Bioinformatics **18**: 335–6.

107 KUCHINSKY, A., K. GRAHAM, D. MO, A. ADLER, K. BARBARIA AND M. CREECH. 2002. Biological storytelling: a software tool for biological information organization based upon narrative structure. Presented at Advanced Visual Interfaces ‘02, ACM SIGGROUP Bulletin **23(2)**: 4–5.

108 KURTZ, S., A. PHILLIPPY, A. L. DELCHER, M. SMOOT, M. SHUMWAY, C. ANTONESCU AND S. L. SALZBERG. 2004. Versatile and open software for comparing long genomes. Genome Biol. **5**: R12.

109 KVASNICKA, H. M. AND J. THIELE. 1995. [3-dimensional reconstruction of serial sections in light microscopy]. Pathologe **16**: 128–38.

110 LAKOWICZ, J. R. 1999. *Principles of Fluorescence Microscopy*, 2 edn. Kluwer Academic/Plenum, New York, NY.

111 LAKOWICZ, J. R., I. I. GRYCZYNSKI AND Z. GRYCZYNSKI. 1999. High throughput screening with multiphoton excitation. J. Biomol. Screen. **4**: 355–62.

112 LAKOWICZ, J. R., H. SZMACINSKI, K. NOWACZYK, K. W. BERNDT AND M. JOHNSON. 1992. Fluorescence lifetime imaging. Anal. Biochem. **202**: 316–30.

113 LANSFORD, R., G. BEARMAN AND S. E. FRASER. 2001. Resolution of multiple green fluorescent protein color variants and dyes using two-photon microscopy and imaging spectroscopy. J. Biomed. Opt. **6**: 311–8.

114 LEE, J. P., D. CARR, G. GRINSTEIN, J. KINNEY AND J. SAFFER. 2002. The next frontier for bio-and cheminformatics visualization. IEEE Comput. Graphics Appl. **22(5)**: 6–11.

115 LEE, M. S., S. S. PARK AND H. S. PARK. 2003. UniPath: a knowledge representation system for biological pathways. Genome Informatics **14**: 681–2.

116 LEVINE, B. AND B. HERMAN. 1998. Quantitative fluorescence resonance energy transfer measurements using fluorescence microscopy. Biophys. J. **74**: 2702–13.

117 LEWIS, S., S. SEARLE, N. HARRIS, et al. 2002. Apollo: a sequence annotation editor. Genome Biology **3**: RESEARCH0082.1–14.

118 LIPPINCOTT-SCHWARTZ, J. AND G. H. PATTERSON. 2003. Development and use of fluorescent protein markers in living cells. Science **300**: 87–91.

119 LIPPINCOTT-SCHWARTZ, J., E. SNAPP AND A. KENWORTHY. 2001. Studying protein dynamics in living cells. Nat. Rev. Mol. Cell. Biol. **2**: 444–56.

120 LORAINE, A. AND G. HELT. 2002. Visualizing the genome: techniques for presenting human genome data and annotations. BMC Bioinformatics **3**: 19.

121 MANE, K. K. AND K. BORNER. 2004. Mapping topics and topic bursts in PNAS. Proc. Natl Acad. Sci. USA **101**: 5287–90.

122 MAYER, T. U., T. M. KAPOOR, S. J. HAGGARTY, R. W. KING, S. L. SCHREIBER AND T. J. MITCHISON. 1999. Small molecule inhibitor of mitotic spindle bipolarity identified in a phenotype-based screen. Science **286**: 971–4.

123 MCCONNELL, P., K. JOHNSON AND S. LIN. 2002. Applications of Tree-Maps to hierarchical biological data. Bioinformatics **18**: 1278–9.

124 MITCHISON, T. J. 2005. Small-molecule screening and profiling by using automated microscopy. ChemBiochem **6**: 33–9.

125 MOHLER, W., A. C. MILLARD AND P. J. CAMPAGNOLA. 2003. Second harmonic generation imaging of endogenous structural proteins. Methods **29**: 97–109.

126 MONTGOMERY, S. B., T. ASTAKHOVA, M. BILENKY, et al. 2004. Sockeye: a 3D environment for comparative genomics. Genome Res. **14**: 956–62.

127 MORELAND, J., A. GRAMADA, O. BUZKO, Q. ZHANG AND P. BOURNE. 2005. The Molecular Biology Toolkit (MBT): a modular platform for developing molecular visualization applications. BMC Bioinformatics **6**: 21.

128 MUILU, J., PATRICIA-RODRIGUEZ-TOMÉ AND A. ROBINSON. 2001. GBuilder – an application for the visualization and integration of EST cluster data. Genome Res. **11(1)**: 179–84.

129 MUNZNER, T., F. GUIMBRETIÈRE, S. TASIRAN, L. ZHANG AND Y. ZHOU. 2003. TreeJuxtaposer: scalable tree comparisons using focus + context with guaranteed visibility. ACM Trans. Graphics **22**: 453–62.

130 MYERS, C. L., X. CHEN AND O. TROYANASKA. 2005. Visualization-based discovery and analysis of genomic aberrations in microarray data. BMC Bioinformatics **6**: 146.

131 NAGASAKI, M., A. DOI, H. MATSUNO AND A. MIYANO. 2003. Genomic Object Net: I. A platform for modelling and simulating biopathways. Appl. Bioinformatics **2**: 181–4.

132 NORTH, C. AND B. SHNEIDERMAN. 2000. Snap-together visualization: a user interface for coordinating visualizations vice relational schema. Presented at the Conf. on Advanced Visual Interfaces, Palermo, Italy: 128–35.

133 OINN, T., M. ADDIS, J. FERRIS, et al. 2004. Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics **20**: 3045–54.

134 PADDISON, P. J., A. A. CAUDY AND G. J. HANNON. 2002. Stable suppression of gene expression by RNAi in mammalian cells. Proc. Natl Acad. Sci. USA **99**: 1443–8.

135 PARR, C. S., B. LEE, D. CAMPBELL AND B. B. BEDERSON. 2004. Visualizations for taxonomic and phylogenetic trees. Bioinformatics **20**: 2997–3004.

136 PARVIN, B., Q. YANG, G. FONTENAY AND M. H. BARCELLOS-HOFF. 2002. BioSig: an imaging bioinformatic system for studying phenomics. IEEE Comput. **35(7)**: 65–71.

137 PAWLEY, J. E. 1996. *Handbook of Confocal Microscopy*. Plenum Press, New York, NY.

138 PINKEL, D., R. SEGRAVES, D. SUDAR, et al. 1998. High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. Nat. Genet. **20**: 207–11.

139 PLAISANT, C. 2004. The challenge of information visualization evaluation. Presented at the Conf. on Advanced Visual Interfaces, Gallipoli, Italy: 109–16.

140 PLATANI, M., I. GOLDBERG, A. I. LAMOND AND J. R. SWEDLOW. 2002. Cajal Body dynamics and association with chromatin are ATP-dependent. Nat. Cell Biol. **4**: 502–8.

141 PLATANI, M., I. GOLDBERG, J. R. SWEDLOW AND A. I. LAMOND. 2000. *In vivo* analysis of cajal body movement, separation, and joining in live human cells. J. Cell Biol. **151**: 1561–74.

142 PLOTNIKOV, S., V. JUNEJA, A. B. ISAACSON, W. A. MOHLER AND P. J. CAMPAGNOLA. 2005. Optical clearing for improved contrast in second harmonic generation imaging of skeletal muscle. Biophys J. **90**: 328–39.

143 RAMANUJAM, N. 2000. Fluorescence spectroscopy of neoplastic and non-neoplastic tissues. Neoplasia **2**: 89–117.

144 REES, C., J. DEMETER, J. MATESE, D. BOTSTEIN AND G. SHERLOCK. 2004. GeneXplorer: an interactive web application for microarray data visualization and analysis. BMC Bioinformatics **5**: 141.

145 ROST, U. AND E. BORNBERG-BAUER. 2002. TreeWiz: interactive exploration of huge trees. Bioinformatics **18**: 109–14.

146 RUEDEN, C., K. W. ELICEIRI AND J. G. WHITE. 2004. VisBio: a computational tool for visualization of multidimensional biological image data. Traffic **5**: 1–7.

147 RUTHS, D. A., E. S. CHEN AND L. ELLIS. 2000. Arbor 3D: an interactive environment for examining phylogenetic and taxonomic trees in multiple dimensions. Bioinformatics **16**: 1003–9.

148 SACCONI, L., M. D'AMICO, F. VANZI, T. BIAGIOTTI, R. ANTOLINI, M. OLIVOTTO AND F. S. PAVONE. 2005. Second-harmonic generation sensitivity to transmembrane potential in normal and tumor cells. J. Biomed. Opt. **10**: 024014.

149 SALDANHA, A. J. 2004. Java Treeview – extensible visualization of microarray data. Bioinformatics **20**: 3246–8.

150 SARAIYA, P., C. NORTH AND K. DUCA. 2004. Presented at the IEEE Symp. on Information Visualization, Austin, TX, USA: 1–8.

151 SARAIYA, P., C. NORTH AND K. DUCA. 2005. Visualization for biological pathways: requirement analysis, system evaluation, and research agenda. Inf. Visual. **4(3)**: 191–205.

152 SATO, N. AND S. EHIRA. 2003. GenoMap, a circular genome data viewer. Bioinformatics **19**: 1583–4.

153 SCHREIBER, S. 2003. The small-molecule approach to biology. Chem. Eng. News **81**: 51–61.

154 SCHWARTZ, S., L. ELNITSKI, M. LI, et al. 2003. MultiPipMaker and supporting tools: alignments and analysis of multiple genomic DNA sequences. Nucleic Acids Res. **31**: 3518–24.

155 SCHWARTZ, S., Z. ZHANG, K. A. FRAZER, et al. 2000. PipMaker – a web server for aligning two genomic DNA sequences. Genome Res. **10**: 577–86.

156 SEO, J., M. BAKAY, Y.-W. CHEN, S. HILMER, B. SHNEIDERMAN AND E. P.

HOFFMAN. 2004. Interactively optimizing signal-to-noise ratios in expression profiling: project-specific algorithm selection and detection *p*-value weighting in Affymetrix microarrays. Bioinformatics **10**: 2534–44.

157 SEO, J. AND B. SHNEIDERMAN. 2004. A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. Presented at the IEEE Symp. on Information Visualization, Austin, TX, USA: 65–72.

158 SEO, J. AND B. SHNEIDERMAN. 2002. Interactively exploring hierarchical clustering results. IEEE Comput. **35**: 80–6.

159 SHAH, N., O. COURONNE, L. A. PENNACCHIO, et al. 2004. Phylo-VISTA: interactive visualization of multiple DNA sequence alignments. Bioinformatics **20**: 636–43.

160 SHANNON, P., A. MARKIEL, O. OZIER, et al. 2003. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. **13**: 2498–504.

161 SHARAN, R., A. MARON-KATZ AND R. SHAMIR. 2003. CLICK and EXPANDER: a system for clustering and visualizing gene expression data. Bioinformatics **19**: 1787–99.

162 SHAROV, A. A., D. B. DUDEKULA AND M. S. H. KO. 2005. A web-based tool for principal component and significance analysis of microarray data. Bioinformatics **21**: 2548–9.

163 SHNEIDERMAN, B. 1996. A Task by Data Type Taxonomy for Information Visualization. Presented at the IEEE Workshop on Visual Languages, Boulder, CO, USA: 336–43.

164 SHNEIDERMAN, B. AND C. PLAISANT. 2004. *Designing the User Interface*, 4th edn. Pearson Addison-Wesley, Boston, MA.

165 SPELL, R., R. BRADY AND F. DIETRICH. 2003. A visualization tool for biological sequence analysis. Presented at the IEEE Symp. on Information Visualization, Seattle, WA, USA: 28–35.

166 SPELLMAN, P., M. MILLER, J. STEWART, et al. 2002. Design and implementation of microarray gene expression markup language (MAGE-ML). Genome Biol. **3**: RESEARCH0046.1–9.

167 SQUIRRELL, J. M., D. L. WOKOSIN, J. G. WHITE AND B. D. BAVISTER. 1999. Long-term two-photon fluorescence imaging of mammalian embryos without compromising viability. Nat. Biotechnol. **17**: 763–7.

168 STAATS, B., L. QI, M. BEERMAN, H. SICOTTE, L. A. BURDETT, B. PACKER, S. J. CHANOCK AND M. YEAGER. 2005. Genewindow: an interactive tool for visualization of genomic variation. Nat. Genet. **37**: 109–10.

169 STEIN, L. D., C. MUNGALL, S. SHU, et al. 2002. The Generic Genome Browser: a building block for a model organism system database. Genome Res. **12**: 1599–610.

170 STEVENS, R., C. GOBLE, P. BAKER AND A. BRASS. 2001. A classification of tasks in bioinformatics. Bioinformatics **17**: 180–8.

171 STOTHARD, P. AND D. S. WISHART. 2005. Circular genome visualization and exploration using CGView. Bioinformatics **21**: 537–9.

172 STRYER, L. AND R. HAUGLAND. 1967. Energy transfer: a spectroscopic ruler. Proc. Natl. Acad. Sci. USA **58**: 719–26.

173 STURN, A., J. QUACKENBUSH AND Z. TRAJANOSKI. 2002. Genesis: cluster analysis of microarray data. Bioinformatics **18**: 207–8.

174 SUN, H. AND R. V. DAVULURI. 2004. Java-based application framework for visualization of gene regulatory region annotations. Bioinformatics **20**: 727–34.

175 SWEDLOW, J. R., I. GOLDBERG, E. BRAUNER AND P. K. SORGER. 2003. Informatics and quantitative analysis in biological imaging. Science **300**: 100–2.

176 TABARA, H., A. GRISHOK AND C. C. MELLO. 1998. RNAi in *C. elegans*: soaking in the genome sequence. Science **282**: 430–1.

177 TAMAYO, P., D. SLONIM, J. MESIROV, Q. ZHU, S. KITAREEWAN, E. DMITROVSKY, E. S. LANDER AND T. R. GOLUB. 1999. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. Proc. Natl Acad. Sci. USA **96**: 2907–12.

178 TAO, Y., C. FRIEDMAN AND Y. A. LUSSIER. 2005. Visualizing information across multidimensional post-genomic structured and textual databases. Bioinformatics **21**: 1659–67.

179 TAVERNARAKIS, N., S. L. WANG, M. DOROVKOV, A. RYAZANOV AND M. DRISCOLL. 2000. Heritable and inducible genetic interference by double-stranded RNA encoded by transgenes. Nat. Genet **24**: 180–3.

180 TEBBUTT., S. J., I. V. OPUSHNYEV, B. W. TRIPP, A. M. KASSAMALI, W. L. ALEXANDER AND M. I. ANDERSEN. 2005. SNP Chart: an integrated platform for visualization and interpretation of microarray genotyping data. Bioinformatics **21**: 124–7.

181 THEILHABER, J., A. ULYANOV, A. MALANTHARA, et al. 2004. GECKO: a complete large-scale gene analysis platform. BMC Bioinformatics **5**: 195.

182 THOMAS, C., P. DEVRIES, J. HARDIN AND J. WHITE. 1996. Four-dimensional imaging: computer visualization of 3D movements in living specimens. Science **273**: 603–7.

183 TIMMONS, L., D. L. COURT AND A. FIRE. 2001. Ingestion of bacterially expressed dsRNAs can produce specific and potent genetic interference in *Caenorhabditis elegans*. Gene **263**: 103–12.

184 TORONEN, P., M. KOLEHMAINEN, G. WONG AND E. CASTREN. 1999. Analysis of gene expression data using self-organizing maps. FEBS Lett. **451**: 142–6.

185 TOYODA, T., Y. MOCHIZUKI AND A. KONAGAYA. 2003. GSCope: a clipped fisheye viewer effective for highly complicated biomolecular network graphs. Bioinformatics **19**: 437–8.

186 TOYODA, T. AND A. WADA. 2004. Omic space: coordinate-based integration and analysis of genomic phenomic interactions. Bioinformatics **20**: 1759–65.

187 TRUONG, K., A. SAWANO, H. MIZUNO, et al. 2001. FRET-based *in vivo* Ca^{2+} imaging by a new calmodulin–GFP fusion molecule. Nat. Struct. Biol **8**: 1069–73.

188 WANG, H., F. AZUAJE AND N. BLACK. 2002. Improving biomolecular pattern discovery and visualization with hybrid self-adaptive networks. IEEE Trans. Nanobiosci. **1**: 146–66.

189 WANG, J. AND M. M. BARR. 2005. RNA interference in *Caenorhabditis elegans*. Methods Enzymol. **392**: 36–55.

190 WANG, X. F., A. PERIASAMY AND B. HERMAN. 1992. Fluorescence lifetime imaging microscopy (FLIM): instrumentation and applications. Crit. Rev. Anal. Chem. **23**: 365–9.

191 WARE, C. 2004. *Information Visualization: Perception for Design*, 2nd edn. Morgan Kauffman, San Francisco, CA.

192 WHEELER, D. B., S. N. BAILEY, D. A. GUERTIN, A. E. CARPENTER, C. O. HIGGINS AND D. M. SABATINI. 2004. RNAi living-cell microarrays for loss-of-function screens in *Drosophila melanogaster* cells. Nat. Methods **1**: 127–32.

193 WHITE, J., W. AMOS AND M. FORDHAM. 1987. An evaluation of confocal versus conventional imaging of biological structures by fluorescence light microscopy. J. Cell Biol. **105**: 41–8.

194 WHITE, J. G., J. M. SQUIRRELL AND K. W. ELICEIRI. 2001. Applying multiphoton imaging to the study of membrane dynamics in living cells. Traffic **2**: 775–80.

195 WOOLLACOTT, A. J. AND P. B. SIMPSON. 2001. High throughput fluorescence assays for the measurement of mitochondrial activity in intact human neuroblastoma cells. J. Biomol. Screen. **6**: 413–20.

196 YU, J. Y., S. L. DERUITER AND D. L. TURNER. 2002. RNA interference by expression of short-interfering RNAs and hairpin RNAs in mammalian cells. Proc. Natl Acad. Sci. USA **99**: 6047–52.

197 ZACHARIAS, D. A., J. D. VIOLIN, A. C. NEWTON AND R. Y. TSIEN. 2002. Partitioning of lipid-modified monomeric GFPs into membrane microdomains of live cells. Science **296**: 913–6.

198 ZEEBERG, B. R., W. FENG, G. WANG, et al. 2003. GoMiner: a resource for biological interpretation of genomic and proteomic data. Genome Biol. **4**: R28.

199 Zhang, C.-T., R. Zhang and H.-Y. Ou. 2003. The Z curve database: a graphic representation of genome sequences. Bioinformatics **19**: 593–9.

200 Zhang, L., A. Zhang and M. Ramanathan. 2004. VizStruct: exploratory visualization for gene expression profiling. Bioinformatics **20**: 85–92.

201 Zhong, S., L. Tian, C. Li, K.-F. Storch and W. H. Wong. 2004. Comparative Analysis of Gene Sets in the Gene Ontology Space under the Multiple Hypothesis Testing Framework. Presented at the IEEE Conf. on Computational Systems Bioinformatics, Stanford, CA, USA: 425–35.

44
Using Distributed Data and Tools in Bioinformatics Applications

Robert Stevens, Phillip Lord, and Duncan Hull

1 Introduction to Distributed Resources

The advent of the web has been both a great blessing and a bane for bioinformatics. While it has given researchers access to a large store of data and tools with which to analyze these data, this wide access does have its adverse consequences. The use of web technologies enable organizations and individuals to provide and retrieve data cheaply. This obvious blessing is accompanied by the drawback of distribution and autonomy: the need to join together distributed resources that have developed along different lines. To accomplish most bioinformatics analysis, it is necessary to use more than one resource – a genomic DNA sequence is analyzed with a gene prediction program, a hypothetical gene is translated into protein, that protein is searched against a protein data bank for possible homologs and so on. Traditionally, a bioinformatician has done this by manually cutting and pasting material from page to page on the web. Doing this programmatically necessitates networked applications or distributed computing.

This chapter explores the use of distributed and networked computing. Many of the issues these raise, e.g. fault tolerance, network latency, etc., have been discussed in detail elsewhere [27]; here, we consider those issues which are specific to bioinformatics. We take a practical and historical view, describing how bioinformatics has developed in a networked environment, but we then come right up to date by looking into the future of Grid computing. We will describe the lessons both of failed attempts to address networked applications and the success stories that have emerged.

The fundamental requirement for networked applications arises from the distribution of the laboratories working in bioinformatics, which is characteristic of biology as "small science" or a patchwork of "rival city states" [22].

When sequence data began to be generated, as long back as the late 1970s and early 1980s, researchers wished to exchange data for comparison etc. Distribution at that early stage was by means such as tape, disk and, finally, CDs.

Bioinformatics - From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8

Basic advances in infrastructure solved the problem of transferring data. It was the advent of the web which reduced the requirement for specialist knowledge to do so. Bioinformaticians and biologists were early adopters of this technology. Most laboratories had relatively small amounts of data to present and this was often achieved by use of a text editor, and then simple use of active content via the Common Gateway Interface (CGI). The use of common protocols [HyperText Transfer Protocol (HTTP)], a relatively simple language [HyperText Markup Language (HTML)] and simple user-facing client tools ensured that this technology would become widely adopted.

One of the main virtues of the web is its lack of centralized control; this makes it scalable as there is no single point of failure or bottleneck. Both service providers and consumers are highly *autonomous* from each other. Bioinformatics has used this autonomy to the full; data was stored in nonstandard file formats, with each entity structured using an *ad hoc* ASCII encoding defined by the data holder. This result in a plethora of different file formats; the classic example in bioinformatics is the many formats for biological sequences such as nucleotide and proteins. Thus, for example, the program from EMBOSS `seqret` (http://emboss.sourceforge.net/apps/seqret.html) can take 28 different sequence formats as input and transform them to a format required by subsequent analysis tools.

Many of these resources also grew at the time before relational databases were widespread and scalable. A relational database management system (RDBMS) does provide better data management and query facilities than flat-file delivery, but at the time they were often expensive and lacked the ability to efficiently deal with the large amounts of nonscalar data necessary in biological databases. Even though many bioinformatics resources now use RDBMs, the exchange format is still largely in the legacy flat file.

Tools working over these data have developed in a similar autonomous manner. They take data in a particular format and deliver their results in a nonstandard format. Again, this is usually as some formatted ASCII text. This *heterogeneity* is one of the major hurdles in distributed computing within bioinformatics.

As with the web, this large quantity of data available to the bioinformatician is only really usable because of the use of massive hyperlinking, e.g. the current release of UniProt maintains links to over 60 other resources by cross-linking accession numbers and identifiers from each of these resources. These are usually translated into HTML links when presented to users within a browser.

There are now very many data sources and tools available. The 2005 edition of the annual database issue of *Nucleic Acids Research* describes over 700 [10] data resources, along with hundreds of analysis tools. As new biological techniques develop, such as transcriptomics and proteomics, more resources are developed to accommodate their data. Combined with the desire to perform more complex analyzes, the traditional cut and paste methods of bioinformatics have become increasingly unusable.

Bioinformaticians, therefore, began to develop programs to automate the process, gathering data and tools distributed around the network into single applications. In doing so, they had to overcome a series of problems, and it is these problems and their solutions that are the subject of this chapter. The subjects to be covered are:

- Overcoming the problems of heterogeneity between tools and resources (Section 2).
- the inclusion of data and tools hosted on another computer into an application running on a local computer (Section 4).
- Technologies for this inclusion, such as Common Object Request Broker Architecture (CORBA) and web services (Sections 4.1 and 4.3).

As we review these issues, many examples of distributed or networked applications will arise. In Section 5, we will review some case studies in networked applications in bioinformatics. We finally conclude by introducing cutting edge technologies such as the Grid. This chapter will explain the central role that distributed computing has in modern bioinformatics and provide the reader with the foundations of the field.

2 Heterogeneiety in Bioinformatics Resources

If all distributed resources worked with one standard data format and application programme interfaces (APIs), the existence of the Internet would be enough to deal with most of the problems manifest in distributed computing. The common communication protocols would enable applications to access distributed components running on a variety of machines and their various operating systems. Sadly this is not the case, and considerable difficulties are caused by their differing APIs, query languages, syntaxes, format differents and, finally, but not least, their differing semantics for their content.

This heterogeneiety is the major barrier to distributed computing. Although there is a common communication protocol through the Internet, networked applications still have to deal with the following heterogeneities [18]:

(i) System heterogeneity.

(ii) Syntax heterogeneity.

(iii) Structure heterogeneity.

(iv) Semantic heterogeneity.

System-level heterogeneity means that it is difficult to access and use applications residing on another type of computer and operating system. Recently, quite a number of cross-platform languages have arisen, including Perl and Java, which are widely used in bioinformatics. The use of these languages solves some problems, but introduces others as it often becomes necessary to get these cross-platform languages to interoperate with each other.

CORBA and web services are two technologies used to overcome these low-level problems that have seen usage in the bioinformatics arena. These really are the "plumbing" that connects distributed resources together. These will be discussed further in Section 4.

System level heterogeneity can also impact on the other levels: the line terminator differences between different platforms is really a syntactic problem; it is at this level that we start to find the examples of heterogeneity which affect bioinformatics most heavily. A cynic might observe that one necessary condition for being a bioinformatician is to have written parsers for BLAST output or Swiss-Prot records. The syntactic differences between bioinformatics resources means bioinformaticians are forever writing parsers to convert results into a form suitable for their application. A common syntax would mean that such conversions would be easier to achieve or indeed to understand what the syntax actually means – most bioinformatics flat files have no formal, computable description of their syntax.

Both the autonomy and legacy issues within bioinformatics, however, mean that such common standards are difficult to achieve. ASN.1 [4] is one such attempt at a standard syntax that was not widely adopted. eXtensible Markup Language (XML), which is covered later in Section 4.2, has and is having greater success in bioinformatics.

As well as myriad examples of syntactic heterogeneity, bioinformatics is rife with structural heterogeneiety. Even if we had a common syntax, such as XML, for formatting data, we would still have the problem of how we structure this data within the syntax. A simple case in point is the representation of authors in a literature reference. For example, compare this XML representation of one of the authors from UniProt:

```
<authorList>
   <person name="Lord P."/>
</authorList>
```

with this from GenBank:

```
<Person-id_name>
   <Name-std>
       <Name-std_last>Lord</Name-std_last>
```

```
        <Name-std_initials>P.</Name-std_initials>
    </Name-std>
</Person-id_name>
```

The former makes no attempt to structure the name, while the latter splits initials and last names. To compare these two authors then to see if the name is the same requires a set of rules to produce a normalized form. Communication between distributed, autonomous resources requires the use of the structure of the data in order to access just those parts required for analysis. Obviously, differing structures to hold data means problems for using those data.

Much of system, syntactic and structural heterogeneity would be dealt with by a common type system, as described in Section 3, and the lack of such a common system and its consequences are manifest.

Lastly, and certainly not least, is semantic heterogeneiety. This is the difference in meanings between the representations held in different resources. Karp [12] and Davidson and coworkers [6] identified semantic heterogeneity as one of the major barriers to integrating bioinfformatics applications. Ontologies, covered in Chapter 29, are seen as a technology for capturing the semantics of a domain [26]. They offer a mechanism for creating a shared understanding, via vocabulary terms, for a given domain. A particular data resource either adopts the ontology and changes their data or maps their current reprsentation into the ontology. Both of these can be seen with the Gene Ontology [28]. Swiss-Prot uses GO terms directly and mapped its current keywords into an ontology to achieve *de facto* semantic integration.

Much of the work in distributed computing or networked applications is in overcoming these four levels of heterogeneity. Dealing with heterogeneity is a prerequisite for any kind of data integration, as covered in Chapter 42. In this chapter, we concentrate on the aspects of system and syntactic heterogeneiety. The "upper" two levels of heterogeneiety are the subjects of Chapters 29 and 42.

3 Type Systems in Bioinformatics

In building bioinformatics applications, we are interested in manipulating data, but how does a programme know what operations can be performed on any given piece of data? In computer science, a type or datatype is a name for a set of values with shared properties. A *type system* uses these types to constrain the operations that may be performed on the values of a given type [29]. So, for example, the type *integer* typically describes the set of natural numbers. A type system then enforces that only integer operations such as division and multiplication can be performed on integers, and prevents or warns against nonsensical operations such as substring selection from being performed on integers. Type systems are implemented in many different ways, e.g. as part of a programming language or in a database/schema language. This easily leads to heterogeneiety when different type systems are used. Thus,it is worth examining "type" in a little depth.

Type systems differ in their complexity and expressivity, but they generally have several benefits:

- Type checking and safety – making sure only integer operations are performed on integers or "protein operation" performed upon proteins.
- Optimization – manipulating each type and its operations in the most efficient and effective manner possible on a machine.
- Abstraction – creating a model of a domain through a type system and knowing, for example, that "DNA" and "RNA" are both kinds of "nucleic acid".
- Pipeline construction – using the type system to transfer data between components with predictable outcomes.

The heterogeneity pervasive in bioinformatics means there is no globally accepted type system for describing bioinformatics data. This forms a major barrier against integration of distributed data and applications.

In a programming language, if a variable is declared to be of a given type, then a program can check that declarations are valid and safe. Thus, for example, in a statically typed language, a compiler would check that strings were not assigned to variables of type integer, by type checking the statement:

```
Integer i = "this is a string"
```

The statement assigns the value `this is a string` to a variable *i* of type *Integer*, which will cause problems when integer operations like division and multiplication are performed on that string. In a statically typed language like Java, a *compiler* will warn against these sorts of assignments at compile time. In dynamically typed languages like PERL, this type checking is done at run time by the *interpreter* and may involve automatic type coercion from one type to another.

A type system can also help the programmer to abstract away from the binary digits that the computer deals with when the program is executed. In this sense, a type system is a model of the world, which helps users of programmers to describe the data they are manipulating.

Most type systems allow further abstraction than primitive types like `string`, `integer` and `dates` by allowing user-defined or complex types to be built from primitive ones. So, for example, a bibliography might be modeled in a type system as a list of references, each with authors of type string and publication dates of type date. Again, as well as having better descriptions of the data, it is possible to define the operations performed on those descriptions, as seen in object-orientated programming languages.

Joining the inputs and outputs of two programs together, as seen in networked applications, can be facilitated by a type system. Given an instance of a type, such as a UniProt record, a type system can help to:

- Infer its type by *type inference* – this automatically assigns a type if it is not explicitly stated
- Identify a set of other types into which a UniProt record can be transformed
- Identify a set of processes that accept UniProt records as input. This includes processes that accepts parts of UniProt records, such as protein sequences and database identifiers. This can be useful when constructing programs or workflows that process data as it allows the programmer to find out what they can plug in next, at any given point in the dataflow.
- Identify a set of processes which output UniProt records. This can be useful in planning techniques, which start from an endpoint and work backwards to a starting point [3].

As a consequence of the autonomy of bioinformatics, there is no globally agreed upon and used type system for modeling bioinformatics data. Given the autonomous and distributed nature of bioinformatics, this situation is unlikely to change. Instead of a global type system, we have many different type systems implemented in the various languages used in bioinformatics. This causes heterogeneiety at the system level, as 'integer' may be represented as 2 bytes on one system and 4 bytes in another, meaning operation of the programme may not be the same. Also, complex types such as "protein Data Bank entry" will be different in each application or tool. In addition, we have the type systems of each of the 700+ [10] databases that exist in bioinformatics.

As well as having distributed data that is autonomously produced and therefore heterogeneous, bioinformatics data is often semistructured and weakly typed – the ubiquitous flat file being the best example of this. These flat file entries will often be returned as "string" to a programme. The structure and the type of the data are thus implicit, and usually have to be parsed and transformed to some type system by the application developer.

An important consequence of the lack of a type system in bioinformatics is that joining distributed resources together frequently requires the use of "shims" to align the inputs and outputs of two closely related pieces of data [14]. An example is shown in Figure 1.

These shims perform some of the operations associated with a type system in an *ad hoc* manner. They achieve this by coercing, inferring, casting and

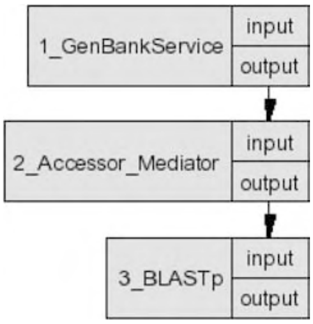


Figure 1 A shimming scenario in bioinformatics. A GenBankService (1) producing a GenBank record needs to be plugged into a BLASTp service (3), which accepts a protein sequence. Getting services (1) and (3) to interoperate requires an Accessor_mediator service (2) to extract the protein sequence from the GenBank record.

constructing types when they process the weakly typed and semistructured data that is common in bioinformatics.

A good type system is very helpful in manipulating data. Bioinformatics application builders spend much of their time overcoming the consequences of having no global type system. In the next section, we will see that describing the type of data being moved and dealing with system heterogeneity is one of the major factors in achieving distributed computing.

4 Plumbing Bioinformatics Resources

When developing applications in a distributed environment, a fundamental activity is to join resources, and pass data between data stores and analytical tools. To the programmer attempting to use distributed bioinformatics resources within a single program in this way, the fact that different resources present different programmatic interfaces (or none at all) implies that a considerable amount of effort has to be expended. For example, consider the case when a Java program needs to access a remote database written in C++ and feed some of the results to another program written in Perl. In order to do this the programmer must cope explicitly with the different languages in question – the C++ will be invoked in one way, Perl in another – and with the distribution. The C++ program is likely to be invoked differently if it were to be available on the local machine. This kind of coding requires a large amount of effort and the resultant program tends to be fragile. If it is decided to mirror the database locally, the Java program will need rewriting. If the Perl program is ported to C++, again the Java needs rewriting.

It would be possible to *integrate* all the required tools and databases in a single place – in practice this is often impractical, so instead, interoperability, providing the ability for resources to work together while remaining separate or "loosely coupled" is often more desirable. One solution to this problem is to use some *middleware* technology. As the name suggests, this technology adds a middle architectural layer which abstracts from the different languages, systems and locations. Following the example above, instead of writing code in Java to invoke the C++ database directly, both will be *wrapped* with the middleware technology. This technology then has the task of managing the communication between these wrappers around the two different implementations. While this seems overly complex, it actually simplifies many issues. The Java programmer no longer has to worry either whether the C++ database is local or remote, nor does it matter that interaction is needed with both C++ and Perl. Therefore middleware technologies offer an attractive solution to overcoming system and syntactic heterogeneity in a distributed setting.

In this section, we consider a number of middleware technologies and their use within bioinformatics. First, we describe ww CORBA (Section 4.1). This technology provides both a standard mechanism for data structuring and for providing (relatively) transparent access to the structured data across distributed systems. Second, we describe XML (Section 4.2) which provides a common syntax for structuring data. Finally, we examine web services (Section 4.3), which use HTTP – the protocol on which the web is built – to overcome system level heterogeneity, leaving syntactic problems to XML.

4.1 CORBA

CORBA is one solution to the problems of system and syntax heterogeneity. Its use in bioinformatics arose in the late 1990s and is now a mature industry standard, and has been widely proposed as a solution to the problem of distribution in bioinformatics.

CORBA attempts to present a common view of the world by presenting it from an object-modeling perspective. To continue the example introduced earlier, to the Java program using CORBA, both the C++ database and the Perl program would appear to be Java objects. Interaction with these objects would be identical to interaction with any other Java object. Similarly, on the C++ side, queries to the database would appear to be coming from a local C++ object rather than from a remote Java program.

To enable this technology, the target resources can be described in a common language. This common language can then be compiled automatically into the programming language of choice, which then enables these target resources to appear as if they are part of the local host application. A core feature of the CORBA specification is this language – the Interface Definition Language

(IDL). This language is used to describe the operations, including return types and arguments taken, that the target resources perform and it can be used by CORBA-compliant tools to generate code for both providing access to the services and the means for the services to be accessed.

As the name suggests, IDL is a language that describes an object's interface; it does not, however, describe how the behavior offered by that interface is implemented. Thus, it is independent of any individual programming language, although, in practice, it looks somewhat like C++.

The actual work of enabling this interaction between different IDL-described resources is a fairly complex procedure. By analogy to financial brokers who organize complex financial transactions, requests between CORBA objects are mediated by an object request broker (ORB). The CORBA bus is the communication system between objects used by an ORB. Different vendors supply their own ORBs, so one person's client may, in theory, use a different ORB to the one being used by the server with which they wish to communicate. This is the importance of the common part of CORBA – it is a common architecture for ORBs. One of the specifications in the CORBA standard is that ORBs will interoperate. This means that one vendor's ORB will (hopefully!) work seamlessly with another vendor's ORB. Figure 2 shows the relationship between client and server objects, their ORBs and the CORBA bus along which they communicate.

While the interaction between ORBs appears to be precisely specified, there is actually considerable scope for variation between them. For most CORBA applications, performance is constrained by network latency – the amount of time it takes to make an initial connection. Theoretically, therefore, good CORBA implementations should all have a similar performance bound by the network; however, in practice, it can vary significantly.

Although CORBAs middleware technology offers a solution to system and syntax heterogeneity, it does not of itself provide a solution to structure heterogeneity. There is nothing to prevent different resource providers modeling objects in different ways. Again, considering the earlier example, the database may return objects representing authors of bibliographic references, while the Perl program may consume similar, but different, structured objects. The Java programmer must therefore map between the different object representations.

The CORBA technology and specifications are overseen by the Object Management Group (OMG). In addition to the basic specifications, the OMG also defines a complex social process for standardizing the IDL representations for a specific domain.

Many of the standards produced by the OMG are specifications for services which support the use of distributed objects. When, for example, objects are distributed, they have to be found by programs which wish to use them. Thus, the two fundamental services offered are the *naming service*, that finds objects

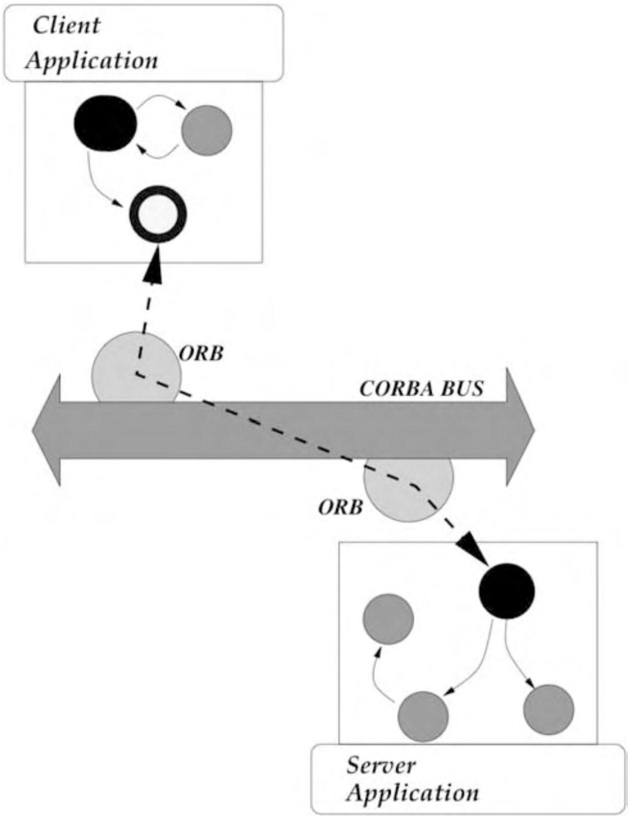


Figure 2 The CORBA bus.

by name, and the *trading service*, that finds objects by the requests they answer. Other services include object life cycle services, event services and security services. Many other services are specified within the CORBA specifications, but they all support the use of distributed objects and make use of the CORBA bus to communicate.

In addition to these generic specifications, there are also a number of bioinformatics domain specifications. The European Bioinformatics Institute (EBI), in particular, invested significant effort in providing CORBA solutions for many services, including EMBL (http://corba.industry.ebi.ac.uk). The OMG formed the Life Sciences Research Group (LSR) that has developed several standards for services including bibliography and sequence resources [5, 19, 20]. However, the uptake of CORBA by the community has not been widespread. The main reasons for this have been the perception that CORBA is too heavyweight a mechanism – the large effort required to develop the standards seen as necessary by the OMG and the implementations themselves obstructed development. Many of the early ORBs were expensive and focused on enterprise-level computing, which did not fit well with the bioinformatics cottage industry. In addition, many ORBs themselves did not actually interoperate. Finally, CORBA seemed to be plagued by continual problems with tunneling through firewalls, defeating the promise of location independence.

Despite these difficulties, CORBA now offers a usable technology. Many languages, such as Java, now provide basic ORBs which can be used for free. If a fine-grained, object-based, platform- and language-independent middleware is required, CORBA is probably still the best option.

4.2 XML in Bioinformatics

Syntactic heterogeneity is commonplace within bioinformatics. This situation is made worse by the heavy use of "flat file" formats. Many of these have no formal specification – an exact description of what is and is not allowed within a file – meaning that parsers are both hard to write and standard tools, such as lex and yacc [13], are not able to automate the process. This also creates a huge versioning problem – when the format of a BLAST record, for instance, changes there is nothing that tells you it has changed until all the software, which depend on the format, breaks.

One standard syntax which is in common use is HTML. This does a reasonably good job of describing documents so that they can be presented on the web. However, this presentational format does not reveal the underlying structure of the data, such as all the fields, etc., in a flat-file database entry. In fact, it can make the situation worse. As quite a few bioinformatics resources are available only through a web site, automated use can require "screen scraping" techniques, i.e. using a program to make an interpretation of HTML which is meant for presentation to humans. This works but is even more fragile than flat-file technology. Every time the web site presentation changes, it may potentially break downstream programs.

XML (http://www.w3.org/xml) is a potential solution to this problem. Despite its name, it is not actually a markup language, but rather a description of how to specify new markup languages or "application" of XML, such as HTML. (This is not actually true. HTML is actually an application of SGML, which is a somewhat more complex fore-runner of XML and HTML does not quite conform to the restrictions of XML. There is a new version of HTML, called XHTML, which is valid XML.) Languages such as the Standardized General Markup Language (SGML) are "languages for languages" or metalanguages. It can be used to define the structure of documents in a particular domain. So, HTML describes the structure of a web page to have a header and a body that may contain headings, paragraphs, tables, images, etc. With SGML, it is possible to define a formal grammar which specifies which tags are valid within a document and how these tags relate to each other. The meta-language is used to define what elements may appear in a document, where they can appear and how often, etc. This grammar provides a schema for the document – XML Schema.

XML is a lighter weight version of SGML, but it has a language for describing document structure – hence the term "extensible". HTML, on the other hand, is a document structure for a particular domain (presentation on the web) and is not itself extensible to another domain. XML does, however, provide a common syntax in the now familiar tree of nested elements (see the description of names in Section 3 where *elements* are enclosed in angle brackets, etc. This combination of document structure, common syntax and available tools make XML a powerful tool. It is possible, for instance, to check that a document is *valid* XML syntax – all elements are matched with closing elements and elements are properly nested. etc. Tools are also provided that take a document and together with its XML schema not only validate the document, but *verify* it against the schema. Any part of the document that contravenes the rules in the schema means a document will not verify.

Uptake of XML within the sciences has been widespread; the Chemical Markup Language (CML) is widely recognized as one of the first substantial uses of XML technologies (http://www.xml-cml.org/information/ position.html).

When it was first introduced in 1996, initial uptake of XML in bioinformatics was relatively slow, due to rapidly changing specifications and poor tool support. XML is now, however, relatively mature. There are standard parsers, APIs for the Document Object Model (DOM) and simple API for XML (SAX), cross-linking and query standards (XPointer, Xquery) and stylesheet and transformation languages (CSS and XSLT) (http://www.w3.org/xml). Another large issue has been the substantial investment in legacy tools; although both UniProt records (http://www.ebi.uniprot.org/support/documents.shtml) and National Center for Biotechnology Information (NCBI) BLAST results (http://xml.coverpages.org/ncbiDataModel.html) have been available as XML for a number of years, most tools still work on the original flat-file formats.

It is notable, however, that for newer databases, such as Interpro (http://www.ebi.ac.uk/interpro/), XML is increasingly becoming the representation of choice. It is even more widely used as an interchange language, often as a layer on top of a relational database, with languages such as Bioinformatics Sequence Markup Language (BSML) (http://www.visualgenomics.com/products/) that is mostly aimed at describing sequence features rather than sequences *per se*.

While XML has an important role to play in distributed resource management, it is important to understand what it does and does not provide. It does offer a standard syntax and language for describing and customizing the use of this syntax, i.e. it is extensible. It supplies a type system to the data it describes. It does, however, provide a "self describing" format. The presence of an <AUTHOR/> tag into a document is, intrinsically, no more informative than <H1/>. For an XML format to be useful, therefore, it must be agreed on by the entire community and the semantics, or meaning, of the individual tags clearly defined. This degree of centralization and control, in many ways, contradicts the distribution and decentralization which is characteristic of bioinformatics.

XML is not a universal solution to the problems of distributed data integration [1], but by overcoming syntactic heterogeneity it does provide valuable facilities for describing data structure which most bioinformaticians are likely to use at some point.

4.3 Web Services

While CORBA proved technically successful, in recent years web services (http://www.w3.org/TR/wsdl) have started to take over as the predominant paradigm for client–server interactions. Perhaps, partly, in response to the perceived complexity of CORBA, web services were designed to build on the success of the world wide web and one of its key messages, that many things can be achieved with a simple system. Since the initial advent of web services a panoply of toolkits and specifications have arisen, provided a fully featured and functional technology.

Web services take the same basic approach to distribution as CORBA, but with several significant differences. At heart, both CORBA and web services take a description of a service being offered and produce code for developing clients and servers. Web services takes the view that distributed tools and data are offered as "services" to applications that wish to use them. Any such *service-orientated architecture* has the following requirements:

- A standard communication protocol between services and host application.
- A uniform data representation and exchange mechanism.
- A standard language to describe the service's attributes and operations.
- A mechanism to register and discover web services.

The web services technology stack fulfills these goals. The Simple Object Access Protocol (SOAP) is the channel used for communication between a web services provider application and a client application. SOAP re-uses the HTTP for transporting messages. Messages are passed between services using XML documents. The structure for SOAP message includes an envelope and a body. The body itself describes a message to a service, e.g. a call to a particular operation or communicate failure. The envelope gives the metadata necessary for this invocation

As CORBA's IDL is used to describe services, web services use another XML document type to describe services. The Web Services Definition Language (WSDL) is used to describe the operations and attributes for a service. A WSDL description, like CORBA's IDL, is used to generate both client and server code for the service. WSDL can be used to generate a variety of programming languages on a variety of platforms. Client and server, once deployed, are ready to pass SOAP messages between one another.

Finally, just like CORBA's naming and trading services, web services need to be discovered for use. WSDL documents can be placed in a registry based on the Universal Description, Discovery and Integration (UDDI) (http://www.uddi.org) framework, and these registries can be searched to retrieve WSDL descriptions of interest. A user would then compile that file to generate a client and use information from the WSDL description to locate and use the service.

Web services take a different approach to that of CORBA. While the latter uses a remote object approach which provides the ability for passing around data and subsequent fine grained client–server interaction, with that data, web services use a "document-based" paradigm. Here, potentially complex structured data is passed between services in bulk. The hope is that instead of a series of fine-grained interactions between client and server, fewer coarser, but richer, interactions will happen – something of clear benefit when faced with any serious problems of network latency or failure.

Ironically enough, this document based, service-orientated architecture bears many similarities to the traditional bioinformatics approach of passing complex data from Perl-driven CGI scripts. However, standard technologies for describing service interfaces, for presenting these services, for structuring the documents passed between services and for discovering services, were all lacking in bioinformatics' original *ad hoc* implementation of this service-orientated architecture, and are all provided by the web services technologies.

Both web services and CORBA provide technical solutions to integration and both have been deployed in bioinformatics (see Chapter 42). Web services have had the greater uptake, due to their perceived lighter approach and ability to do bulk transfer.

There have been a number of different projects, some of which are described in Section 5, which use some or all of the web services technology stack. In addition to these projects, quite a few service providers are now providing programmatic interfaces as a web service, including for example Interpro (see http://www.mygrid.org.uk for a list of available web services).

Although bioinformaticians are using web services, there are a number of peculiarities in their usage within the domain. The intention of the design behind web services was that all the data would be modeled as XML. In bioinformatics, this is a big requirement to place on service providers, as presentation of many of the standard bioinformatics data types in XML is nontrivial. In addition, there is a large legacy with which to deal. For this reason, many of the web services use XML only as far as describing their complex, structured, flat-file results as `xsd:string` – the XML string type. In short, these web services look much like their CGI progenitors, but with a thin web services veneer.

It seems likely that the uptake of web services will continue, as a result of several synergistic advances: the web services specifications and toolkits are stabilizing, standard XML representations for bioinformatics are becoming more widespread, and, finally, client tools are becoming available.

5 Case Studies in Distributed Bioinformatics

In this section, we consider a number of different projects as case studies for the use of distributed technologies within bioinformatics. All of these projects have focused on providing a middleware layer to either integrate or inter-operate over heterogeneous data sources and to provide some degree of transparency to the distribution of the resources. Chapter 42 also describes examples of integration of distributed resources in bioinformatics.

5.1 ISYS

ISYS is described as a decentralized, integration system, as opposed to a distributed system *per se* [21]. The aim was to enable the integration of a number of different components, to allow their interoperation without a tight entanglement of their respective code bases. ISYS was particularly directed at producing user interface-driven interoperation, enabling data to be passed freely between components. ISYS achieves this with several key components:

- The ClientBus. The various components to be integrated can be registered with the ClientBus. All communication between components then passes through this system.
- An event model. The ClientBus is used to transfer Events between different components. Components can produce or consume events of different types. As ISYS is aimed at user-driven integration, these event types are quite fine grained, describing, for example, selection or deselection of items within the user interface.

- A domain model. In order to enable different components to communicate a common data model was defined using Java interfaces. Examples of the kinds of data include ISYSSequence (for nucleotide sequence) or IsysTaxon.

While ISYS was not necessarily intended as a distributed system its loose coupling does enable it to be used in this way. Distributed services can be accessed by providing Java wrappers which then call out to these services. For example, Access to the NCBI BLAST service was provided in this way.

5.2 BioMOBY

BioMOBY is an architecture for the discovery and distribution of biological data through decentralized web services [30]. The BioMOBY project has a dual development track with different architectures, known as MOBY-S and S-MOBY (http://www.biomoby.org).

The project originated in the model organism communities where each community had developed standards for sharing data within their closed groups. Many biologists, however, now wish to ask questions requiring data from many different organisms. BioMOBY [30, 31] is an architecture for the discovery and distribution of biological data through web services. As of November 2005 it provided around 400 services for analyzing genomic data. A key feature of BioMOBY is the central registry of services, called "MOBY central", that allows different client programs (e.g. http://mobycentral.icapture.ubc.ca/cgi-bin/gbrowse_moby) to search, browse and execute services that produce or consume a given piece of data, e.g. a GenBank accession number.

5.2.1 MOBY-S

The core philosophy behind MOBY-S has been to ensure "simplicity and familiarity"; the second of these must be emphasized as MOBY-S has focused on familiarity to the bioinformatician. In short, MOBY-S has sought to minimally modify the way that bioinformaticians already work, which largely consists of web-delivered services and CGI scripts. S-MOBY adds to this in three key ways:

(i) A domain ontology. This describes the basic data types in bioinformatics and the relationships between them. In keeping with the notion of familiarity, the ontology is defined as a directed acyclic graph, a representation popularized by the GO [28] (see Chapter 29).

(ii) Messaging. An XML messaging layer has been defined. This provides a description of the wrapped data in terms of the domain ontology, as well as defining some additional relationships between the wrapped data and the information upon which it depends.

(iii) A central discovery server. In addition to the use of the ontology to describe the data provided by the services, the services themselves can be defined in terms of this ontology. This then enables the population of "Moby Central" using the ontology to describe the services.

In some ways, MOBY-S resembles web services architectures, although it predates the widespread adoption of these technologies. Mostly for this reason it does not use all of the web services technology stack.

One of the key decisions made by MOBY-S was to design for an open and extensible system. Moby Central itself is freely accessible. The ontology is accessible and extensible by service providers. MOBY-S itself does not impose any standard for representing the data that it passes around, instead leaving the structuring to the service providers and consumers. This open structure has meant that the original notion of a single Moby Central repository has become outdated; a number of different "in-house" repositories now exist.

5.2.2 **S-MOBY**

The S-MOBY project has been aimed at building on two key architectures. The first of these is the REST architecture [8] and the second is the Semantic Web [2].

As with MOBY-S, the basic idea is to produce middleware enabling the presentation of services with a defined interface and then to augment this with descriptions of these services defined in terms of an domain ontology. Unlike MOBY-S, however, S-MOBY adopted a Resource Description Framework (RDF) (http://www.w3.org/rdf/) representation for its messaging layer with the intention of enabling much deeper structuring of the core bioinformatics datatypes. Secondly with the adoption of OWL-DL (see Chapter 29) as the main formalism for representation of the domain ontology, it should be possible to enable more distribution in the development of the ontology, i.e. different users of S-MOBY should be able to define their own ontology for their own purposes. Finally, central registration of services was never envisaged to be mandatory; descriptions would be published in a standard format on the web, to enable the use of different search engines.

5.3 The Grid Future – the ^{my}Grid Project

The term "Grid" was coined in the mid-1990s, and is the focus of much attention as a distributed computing infrastructure for advanced science and engineering [9]. At its heart, the Grid is about sharing and the transient creation of *virtual organizations* of these shared entities [9]. If this were to happen, some of the same problems, but with different manifestations, are encountered as with data distribution:

- Accessing computational power of differing platforms.
- Scheduling usage across multiple resources.
- Moving data and resources to these platforms.
- Coping with issues of security and authorization.

This access to general resource sharing makes a Grid – defined as "flexible, secure, coordinated resource sharing among dynamic collections of individuals, institutions, and resources" [9]. It is easy to see how this paradigm fits into the world of bioinformatics: A sophisticated, complex bioinformatics *in silico* experiment may involve people, many forms of data, instruments, etc., and all of these could share resources in a Grid [11].

5.4 The ^{my}Grid Project

An example of Grid development is ^{my}Grid (http://www.mygrid.org.uk), a UK e-Science pilot project which is producing Grid middleware infrastructure specifically to support *in silico* experiments in biology [24, 25].

From the issues facing a scientist performing *in silico* experiments come a strong set of requirements to automate the experimental process, its repetition and to support the management of the results. ^{my}Grid addresses these requirements by regarding *in silico* experiments as *workflows* [25]. These workflows automate experiments by orchestrating the services that process data. ^{my}Grid not only supports the creation of the experimental protocol (the workflow), but also the management of the inputs, outputs, intermediates, hypotheses and findings – for the individual and wider groups of scientists. This includes an awareness of the experiments and data holdings of the user, his or her colleagues and the wider scientific community. The aim is to place the scientist at the center of a virtual bioinformatics organization and the flexibility of data management that affords that scientist a *personalized* view of his or her experiment holdings.

Each part of ^{my}Grid addresses the requirements of the research scientist, either by automating tasks or supporting the overall management of experiments and their results.

Service provision. To allow automated interaction with applications during the *in silico* experiment we must provide programmatic access to those applications. this is achieved by making each bioinformatics application available as a web service (see Section 4.3).

Writing the workflows. Automating the experimental process requires an explicit representation of that process sufficient for a computer to execute. A workflow represents a procedure, such as a bioinformatics analysis, as a set of processes and the relationships between those processes (see Chapter 42). It is the level of abstraction that is an important aspect of workflows – the user has a high-level rather than "assembly-level" access to the analysis. Thus, the user describes what he or she wishes to accomplish, not how to accomplish the goal. The ^{my}Grid team have developed the Simple Conceptual Unified Flow Language (Scufl) and an application to edit workflows (Taverna Scufl Workbench) [17] to achieve this abstraction. Thus, a biologist or bioinformatician does not need to write a large, bespoke application, but to simply describe what needs to be done and the order in which it is to be done.

Running the workflows. A workflow enactment engine, Freefluo, has been developed for the enactment of workflows written in Scufl [16]. The engine automatically calls each service in the appropriate order and passes data between services. For

Collating the results. Both final and intermediate results from running the workflow are saved either in the users' local file system or ^{my}Grid Information Repository (mIR). A major requirement is to not only automate the experimental process, but also assist the scientist in recording the origin or *provenance* of the large set of interrelated result files.

Automated provenance recording. The workflow environment has been built to automatically generate two kinds of metadata. The first is called *process provenance* and is analogous to a log – recording which services were used to generate the data. The second provides relationships *between data*. In most cases these are dependent on specific services, e.g. a BLAST service will provide a report which has "similar sequences to" the input query sequence. Therefore, each step of the workflow can be annotated with a provenance template which describes the relationship between the data flowing in and out of the process [32]. The recording of the provenance of actions taken in "joining" services, together with the storage of intermediate results, means that each run of an experiment can be fully validated by a user "tracing" back through the coordinated set of results.

Viewing the results. As much of the information has been recorded by machine it must be rendered in a human-readable form. The amount and complexity of the information also require that it must be provided in filtered views that help answer specific questions clearly.

All ^{my}Grid services can be seen in the ^{my}Grid services stack shown in Figure 3. This shows the idea of middleware appearing again: we have web services representing the distributed, heterogeneous resources. Various services in layers on top of those resources provide a collection of APIs against which a programmer can build applications. the main ^{my}Grid application is Taverna – a workflow workbench [17]. Through these technologies ^{my}Grid creates a virtual organization of bioinformatics resources in order to perform *in silico* experiments. It is a flexible tool that allows experiments to be specified, modified and shared. Experiments can be run repeatedly and reliably.

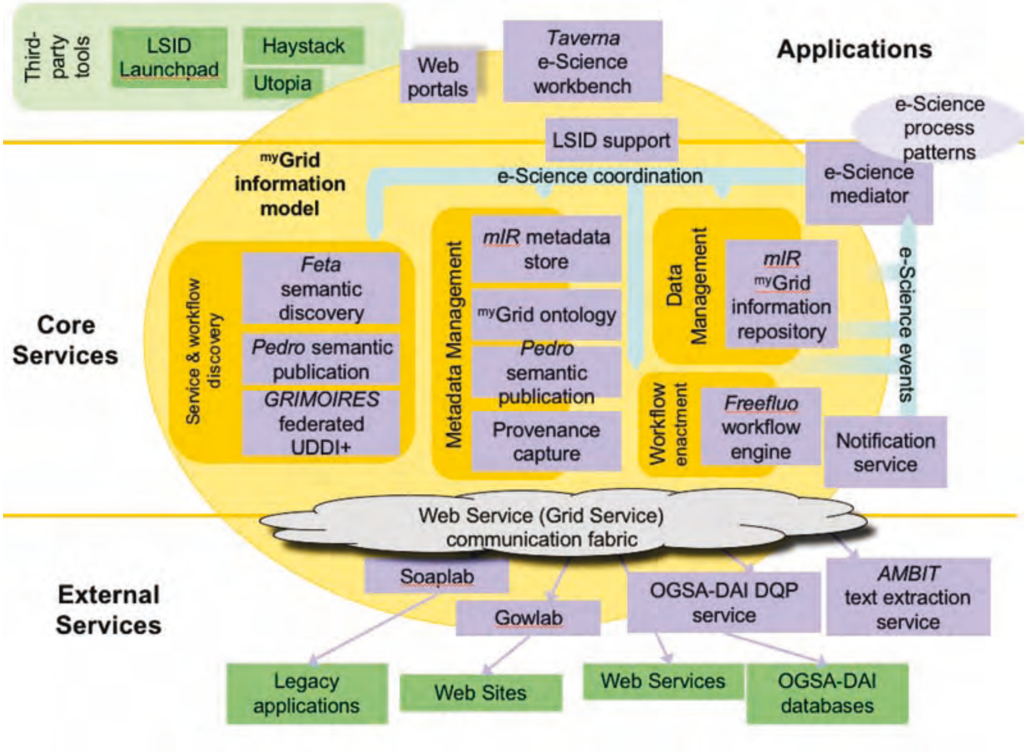


Figure 3 the ^{my}Grid technology stack.

Perhaps, this latter point appears no different from the bespoke solutions so often used in bioinformatics. However, the use of standard web services technologies ease the development of a workflow, compared to more bespoke solutions. This is particularly true as additional services become available, increase the scope of analyses which can be performed.

^{my}Grid is not the only service-orientated architecture solution working within this Grid paradigm. BIRN [15], PathPort [7], the cancer Biomedical Informatics Grid (caBIG) (https://cabig.nci.nih.gov/) and the North Carolina bioGrid (http://www.ncbiogrid.org/) are also using similar solutions within a life science setting.

6 Discussion

In this chapter, we have explored distributed computing within bioinformatics. Starting with a collection of autonomous groups producing data, bioinformatics has a legacy of distributed, massively heterogeneous data and tools. Modern bioinformatics applications have to take this situation into account by either integrating or interoperating between these heterogeneous, autonomous and istributed resources. There is a great deal of technology now on offer to support networked applications in bioinformatics. In this final section, we will explore the question of whether plumbing is enough to achieve the goals of complex, sophisticated bioinformatics analyzes.

Traditional techniques used to resolve syntactic heterogeneity are the development of resource wrappers and exchange formats. Wrappers can transform the appearance of a resource to the external world and dictate what services are available. CORBA offers a standard mechanism by which object views of resources may be developed [23]. Just as CORBA defines the syntactic view of a resource's services, XML can be used to define the syntax and structure of a resource's data. Again, such a technology works by the adoption of standards, so that similar resources use the same data format. A WSDL document describing the services can take this a stage further and we can see that the system and syntactic levels of heterogeneiety can have solutions.

These technologies, however, only offer a mechanism for *plumbing* resources together. A common structural view of a resource's data does not necessarily mean a common semantic view of the same information. At best, these mechanisms offer only an intuitive semantics. For example, an XML tag called `<sequence/>`, within a sequence database entry might have a common understanding, to humans if not machines, but it is unlikely that a tag `<gene/>` would have such an intuitive understanding. These interoperation technologies do not resolve any of the problems of heterogeneous conceptualizations or term usage, because they do not have a mechanism for describing the meaning or knowledge associated with a term. Again, we are beginning to see efforts to resolve this semantic level of heterogeneiety (see Chapter 29).

That individual resources provide access is not enough. For multi-resource applications to be developed, many resources have to work together seamlessly, to cleanly interoperate. For interoperation to be available, the providers of the common views have to describe their resources in such a manner that part of one resource can be passed to another without the intervention of the host developer. For example, the providers of a sequence similarity search tool have to make available services that accept protein sequences in a common form that can be adopted by providers of protein databanks. The alignments returned would also comply to a common description.

Obviously, such a unified, generic approach needs much effort, cooperation and planning. This presents considerable design problems. We are beginning to see the erly stages of Stein's bioinformatics nation [22], through the adoption of web services, but as yet there is no common currency or even agreement what the common language actually means. The sociology of bioinformatics makes it unlikely that a central authority will impose common type systems, etc., but the open nature of the community and the desire for computational solutions to problems will bring in *de facto* standards.

Acknowledgements

P. L. is currently funded by BBSRC grant BBS/B/17131; Duncan Hull is supported by an EPSRC studentship from the ^{my}Grid project GR/R67743/01, with additional support from the School of Computer Science DTA GR/P03339/01.

References

1 ACHARD, F., G. VAYSSEIX, and E. BARILLOT. 2001. XML, bioinformatics and data integration. Bioinformatics, **17**: 115–25.

2 BERNERS-LEE, T., J. HENDLER, and O. LASSILA.2001. The Semantic Web. Sci. Am. May 2001: 35–43.

3 BLYTHE, J., E. DEELMAN, and Y. GIL. 2004. Automatically composed workflows for grid environments. IEEE Intell. Syst. **19**: 16–23.

4 CCITT. Data Communication Networks – Open Systems Interconnection (OSI) – X.208. 1988. *Specification of Abstract Syntax Notation One (ASN.1)*. Blue Book, Melbourne.

5 COUPAYE, T. 1999. Wrapping SRS with CORBA: from textual data to distributed objects. Bioinformatics **15**: 333–8.

6 DAVIDSON, S., C. OVERTON, and P. BUNEMAN. 1995. Challenges in integrating biological data sources. J. Comput. Biol. **2**: 557–72.

7 ECKART, J. D. and B. W. S. SOBRAL. 2004. A life scientist's gateway to distributed data management and computing: the PathPort/ToolBus framework. OMICS **7**: 79–88.

8 FIELDING, R. T. 2000. *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine, CA.

9 FOSTER, I. 2001. The Anatomy of the Grid: Enabling Scalable Virtual Organizations, Lecture Notes in Computer Science, vol. 2150. Springer, Berlin.

10 GALPERIN, M. Y. 2005. The Molecular Biology Database Collection: 2005 update. Nucleic Acids Res. 33: D5–24.

11 GOBLE, C., S. PETTIFER, R. STEVENS, and C. GREENHALGH. 2003. The Grid: blueprint for a new computing Infrastructure. In: FOSTER, I. and C. KESSELMAN (eds.), Knowledge integration: *in silico* experiments in Bioinformatics. Morgan Kaufmann, San Francisco, CA: 121–34.

12 KARP, P. 1995. A strategy for database interoperation. J. Comput. Biol. **2**: 573–86.

13 LEVINE, J., T. MASON, and D. BROWN. 1992. *lex & yacc*, 2nd edn. O'Reilly, Sebastopol, CA.

14 LORD, P., S. BECHHOFER, M. D. WILKINSON, G. SCHILTZ, D. GESSLER, D. HULL, C. GOBLE, and L. STEIN. 2004. Applying semantic web services to bioinformatics: experiences gained, lessons learnt. Proc. 3rd Int. Semantic Web Conf., Hiroshima, Japan. 2004-11-07.

15 NEWMAN, H. B., M. H. ELLISMAN, and J. A. ORCUTT. 2003. Data-intensive e-science frontier research. Commun. ACM **46**: 68–77.

16 T. OINN, M. ADDIS, J. FERRIS, D. MARVIN, M. GREENWOOD, C. GOBLE, A. WIPAT, P. LI, and T. CARVER. Delivering Web Service Coordination Capability to Users. In S. I. FELDMAN, M. URETSKY, M. NAJORK, and C. E. WILLS (Eds.), Thirteenth International World Wide Web Conference (WWW2004), pages 438–439. ACM Press, New York, USA., May 2004.

17 OINN, T., M. ADDIS, J. FERRIS, et al. 2004. Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics **20**: 3045–54.

18 OUKSEL, A. M. and A. SHETH. 1999. Semantic interoperability in global information systems. ACM SIGMOD Rec. **28**: 5–12.

19 REDASCHI, N., K. KRUSZEWSKA, P. LIJNZAAD, and P. RODRIGUEZ-TOMÉ. 1998. Accessing the EMBL database through CORBA – implementation of a browsing server (EMCORBA v2). In Proc. German Conf. on Bioinformatics. Cologne, Germany.

20 RODRIGUEZ-TOMÉ, P., C. HELGESEN, P. LIJNZAAD, and K. JUNGFER. 1997. ACORBA server for the radiation hybrid database. Proc. 5th Int. Conf. on Intelligent Systems for Molecular Biology, Halkidiki, Greece: 250–3.

21 SIEPEL, A., A. FARMER, A. TOLOPKO, M. ZHUANG, P. MENDES, W. BEAVIS, and B. SOBRAL. 2001. ISYS: a decentralized, component-based approach to the integration of heterogeneous bioinformatics resources. Bioinformatics **17**: 83–94.

22 STEIN, L. 2002. Creating a bioinformatics nation. Nature **417**: 119–20.

23 STEVENS, R. and C. MILLER. 2000. Wrapping and interoperating bioinformatics resources using CORBA. Brief. Bioinform. **1**: 9–21.

24 STEVENS, R., R. MCENTIRE, C. GOBLE, M. GREENWOOD, J. ZHAO, A. WIPAT, and P. LI. 2004. ^{my}Grid and the drug discovery process. Drug Discov. Today BIOSILICO **4**: 140–8.

25 STEVENS, R., H. J. TIPNEY, C. WROE, et al. 2004. Exploring Williams-Beuren syndrome using ^{my}Grid, Bioinformatics **20**: i303–10.

26 R. STEVENS, C. WROE, P. LORD, and C. GOBLE. 2003. Ontologies in bioinformatics. In STAAB, S. and R. STUDER (eds.), *Handbook on Ontologies in Information Systems*. Springer, Berlin: 635–57.

27 TANENBAUM, A. S. and M. V. STEEN. 2001. *Distributed Systems: Principles and Paradigms*. Prentice-Hall, Upper Saddle River, NJ.

28 THE GENE ONTOLOGY CONSORTIUM. Gene Ontology: tool for the unification of biology. Nature Genet. **25**: 25–9.

29 WATT, D. A. 1990. Programming Language Concepts and Paradigms. Prentice-Hall, New York.

30 WILKINSON, M. and M. LINKS. 2002. BioMOBY: an open source biological web services proposal. Brief. Bioinform. **3**: 331–41.

31 WILKINSON, M. D. 2004. BioMOBY – the MOBY-S platform for interoperable data service provision. In GRANT, R. P. (Ed.), *Computational Genomics Theory and Application*. Horizon Bioscience, Norwich, UK.

32 ZHAO, J., C. WROE, C. GOBLE, R. STEVENS, D. QUAN, and M. GREENWOOD. 2004. Using Semantic Web technologies for representing e-science provenance. In Proc. 3rd Int. Semantic Web Conf., Lecture Notes in Computer Science: 92–106. Springer, Berlin.

Part 11 Outlook

45 Future Trends

Thomas Lengauer

1 Introduction

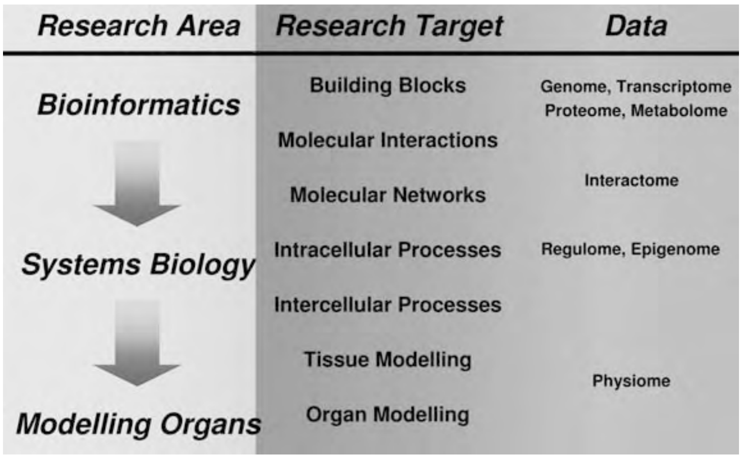
When this book was conceptualized the attempt was made to take into account the major issues pertaining to the computational support of research in molecular biology, pharmaceutics and molecular medicine, and medical practice. The preparation of the book took over 2 years. In a field as dynamic as computational biology it is inevitable that new issues arise that could not be incorporated into the concept of the book at the time of its inception, yet that have gained relevance in the field and are likely to provide considerable impetus to the field in the future.

The goal of this last chapter of the book is to address some of these issues. Some of them are already quite well developed and would have deserved a separate chapter in book; others are emerging and will be ready for separate chapters in potential future editions.

Very likely, even the list of topics addressed in this book is incomplete. I ask for understanding from all those readers that are missing subjects that they feel important – I would like to hear from them!

In general, the development of computational biology follows the direction from the genotype to the phenotype (Figure 1). Most of the activities in the 1990s were targeted at compiling the parts lists of an organism (for different species). The main challenges were to list genes, identify protein sequences and collect protein structures. At the turn of the century a growing number of completely sequenced genomes were available, still mostly from simple unicellular organisms, but the sequences of a few more complex organisms were available (*Caenorhabditis elegans*, *Drosophila melanogaster*) and the human

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8



Research Area	Research Target	Data
Bioinformatics	Building Blocks	Genome, Transcriptome Proteome, Metabolome
	Molecular Interactions	
	Molecular Networks	Interactome
Systems Biology	Intracellular Processes	Regulome, Epigenome
	Intercellular Processes	
	Tissue Modelling	Physiome
Modelling Organs	Organ Modelling	

Figure 1 The development of computational biology.

genome sequence was around the corner. Furthermore, several technologies for transcriptomics had become available, enabling the investigation of the difference between healthy and diseased cell states on the molecular level. Today, the number of completely sequenced genomes reaches well above 300 and the number of ongoing sequencing projects transcends 1700 (http://www.genomesonline.org). Computational biology has entered the stage of uncovering the circuitry of life. The analysis of biochemical networks, be it at the metabolic, gene regulatory or signaling level, can be regarded to be central to present-day bioinformatics efforts. The investigation of protein function in this context with computational biology methods involves a wide variety of biological data, many of which would not have been envisaged 20 years ago. However, much of the research is still concentrated on intracellular processes. The single cell is the universe of today's computational biology. The analysis of intercellular communication and the self-organization of cellular communities has not yet reached the widespread attention of computational biology research. The next levels above the cell – tissues and organs – still await proper computational modeling.

We will now list a number of concrete fields of research that have recently produced exciting research results and that promise to gain importance in the field of computational biology in the future. We will only be able to hint at the results obtained and the potential for the future, but we will point to references that go into more detail on the respective subject.

2 Building Blocks – Post-translational Modification of Proteins

Post-translational modification is a central molecular process for modulating protein structure and function. However, to date, the analysis of protein function based on computational biology methods does not fully incorporate post-translational modification.

There are quite a number of post-translational modifications [69]. The most important ones are phosphorylation and glycosylation. Phosphorylation involves the attachment of phosphate groups (with the catalytic help of a protein kinase) to specific side-chains in proteins, thus modifying their structure and function, e.g. activating or inactivating the protein. Glycosylation adds multimeric sugar molecules to certain side-chains in proteins that modulate the function of the protein, e.g. in cell–cell adhesion, stabilize the protein or serve as an aid in protein folding. Information on post-translational modifications in proteins is collected in special databases [57].

Chapters 28 mentions relevant databases collecting information on post-translational modifications as well as bioinformatics programs and servers that predict sites of post-translational modifications in proteins (see also Ref. [150]). A server that offers several prediction methods for different sites of post-translational modification is available at the Technical University of Denmark (http://www.cbs.dtu.dk/services). Knowing these sites contributes to understanding the function of the respective protein, specifically to resolve its molecular function by manual inspection. In addition, the knowledge of sites of post-translational modification can be used as additional input for statistical learning techniques that use other inputs on the sequence, structure and location of the protein in order to infer protein function automatically [84]. However, today, computational biology has not much surpassed the stage of identifying the sites of post-translational modifications. Modeling of the often very subtle structural consequences is in its infancy.

Protein phosphorylation is at the heart of signaling in eukaryotic cells. Understanding phosphorylation involves studying the phosphorylating enzymes – the protein kinases. The collection of all kinases in an organism is called its kinome. Kinomics is the quest of understanding these proteins and, in this way, gaining a deeper understanding of the process of phosphorylation and its impact on cell signaling. Bioinformatics can help by offering predictions of substrates of kinases or identifying the kinase that is responsible for phosphorylating a particular site [88].

The tremendous structural diversity of multimeric sugars, so-called glycans, has motivated major efforts to understand structure–function relationships of these molecules and their interaction with proteins. The corresponding research area is called *glycomics* [142]. (More specifically, the area is called *glycoproteomics* – as opposed to *glycomics*, which encompasses the study of sugars also independently of proteins. However, often the use of the term *glycomics* is tantamount to *glycoproteomics*.) Several large consortia have formed for glycomics research, such as the US-based Consortium for Functional Glycomics (http://www.functionalglycomics.org/static/consortium), the EurocarbDB consortium funded by the EU (http://www.eurocarbdb.org) and the Japanese Consortium for Glycobiology and Glycotechnology (http://www.jcgg.jp/E/index.html). A recent review mentioning additional projects on glycans is given by Borman and Washington [28].

Glycans are chain-like or tree-like molecules that have no unique three-dimensional (3-D) structure. In contrast to protein–protein interactions that have a digital character, i.e. two proteins bind or they do not bind at any certain time, glycans bind in a more analog fashion, displaying more of a continuum of binding strengths. Singular carbohydrate–protein contacts are rather weak with association constants in the millimolar range. Nature affects biological efficiency by establishing multivalent contacts – a principle that allows for gradually increasing specificity as well as biological control and fine-tuning. Thus, glycans often serve to fine-tune a biological response. A changed glycosylation profile of proteins in a cell or at the cell surface is indicative of many diseases, such that glycomics could possibly aid disease diagnosis in similar ways as transcriptomics.

The major focus in the glycomics initiative is on measurement of glycan structures that are attached to glycosylated proteins and collection of the respective data in databases. However, first steps are also taken in the direction of analyzing these structures with bioinformatics methods [171]. Tools have been developed for extracting glycosylated protein structures from the Protein Data Base [115], modeling the structure of glycans [27] and attaching modeled glycan structures glycosylation sites of structurally resolved proteins [26] (Figure 2). Structure modeling is mainly done with molecular dynamics methods. There is also progress in analyzing the similarities and relationships between different glycans. Tree alignment algorithms have been developed in order to compare different glycans [5] and represent complex patterns in families of glycans [6]. A corresponding server is offered on the internet [7]. There are several servers that offers databases and tools for glycomics [71,114]. An up-to-date list on web links in the glymoics area is available, for instance, at the EurocarbDB web site.

We can expect the computational analysis of post-translational modification of proteins and their impact on protein function to gain in relevance significantly as computational biology matures further.

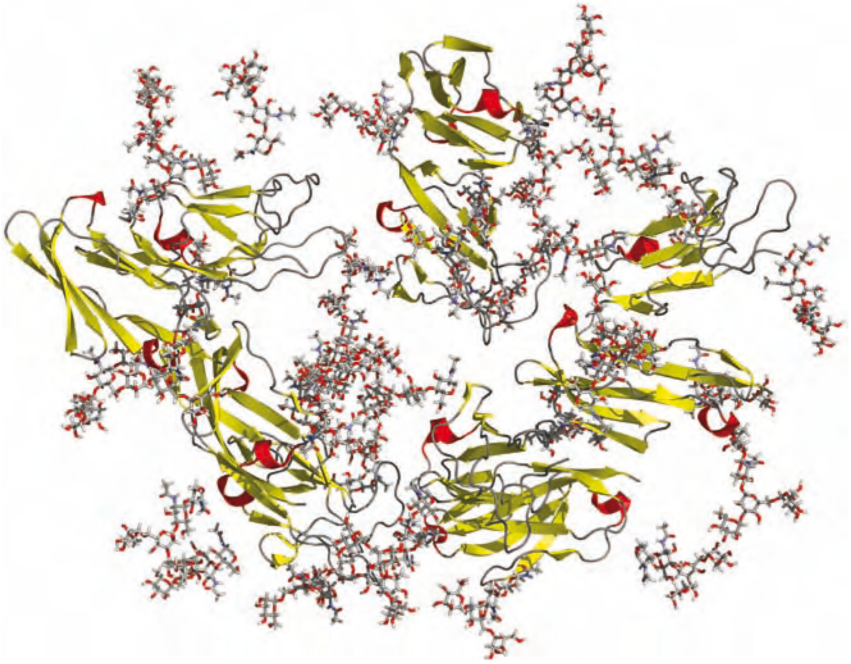


Figure 2 View of a glycosylated immunoglobulin receptor protein (PDB identifier 1 J89). Some of the attached glycans have been taken from the original PDB file; others have been added using the molecular modeling tool GlyProt [26] (courtesy C. W. von der Lieth).

3 Regulation – Synthesis and Degradation Pipeline of RNA and Proteins

Measuring gene and protein expression is such an effective tool for discerning between different cell states that we have devoted several chapters – in fact, all of Part VII of the book – to this topic. Here, measuring gene expression on the level of mRNA transcripts has taken most of our attention. Transcriptomics takes a lopsided view of protein expression, however, because it only considers the first step in a pipeline that is responsible for the final protein expression levels, i.e. transcription. This pipeline has two main stages – mRNA expression regulation and protein expression regulation. In both stages the resulting expression level is the result of production and degradation procedures. Transcriptomics and proteomics measure the resulting balance but, in general, they do not investigate the processes achieving this balance directly. Insight into the regulatory processes achieving this balance is highly desired, however. For instance, problems with degradation processes are presumed to be at the heart of several severe diseases and ageing. Notably, neurodegenerative diseases are caused by aggregating protein products that evade degradation and destroy cells in later stages in life [37]. Manipulating protein degradation is also a target of oncogenic mutations [117]. Protein degradation is also a key step in the destruction of foreign proteins by the immune system (see also Section 8). Furthermore, deeper understanding of regulatory processes in cells will have to involve an explicit account of all steps influencing RNA and protein turnover.

In general, regulation happens at all stages of the pipeline, including mRNA synthesis and degradation, as well as protein synthesis and degradation, and the processes are more complex than originally thought. The findings of the not yet fully understood role of the so-called P-bodies in the regulation of mRNA present only one recent modification of our view of the protein production pipeline [119]. A recent review on the mRNA life trajectory is presented in Ref. [125].

There are a few studies that consider the effects of regulatory processes on protein expression downstream of mRNA at a genome-wide scale. Beyer and coworkers [18] have investigated this issue in yeast. They consider the steps of mRNA expression, protein translation and protein degradation explicitly. They also introduce a computational descriptor of the half-life of a protein. In their analysis they detect significant variation between different cellular compartments and different functional modules. Several case studies points out the essential contribution of protein degradation processes in the regulation of protein expression [91, 179].

There is an additional step that links mRNA and protein expression. This is RNA interference (RNAi), the mechanism, by which small RNA can silence already transcribed genes (or viral RNA). This topic is so important that it deserves a separate subsection.

4 Regulation – RNAi

RNAi [53] has been one of the major biological discoveries of the past decade [130]. RNAi is a mechanism by which small RNA double-stranded sequences with 21 or 22 nucleotides [so-called short interfering RNA (siRNA)] cause the cleavage of, or otherwise inhibit, other RNA (target RNA) in the cell, which is complementary to the siRNA. Target RNA can be viral RNA, which is assumed to be the evolutionary first target of the process. Thus, RNAi is a powerful mechanism of the cell fighting RNA viruses. However, the target RNA can also be mRNA of an expressed gene in the cell. Then, the siRNA effectively silences already transcribed genes. In fact, it is assumed that siRNA has the role of silencing transposable elements (jumping genes) and repetitive genes, in addition to viruses. Subsequently, it has been found that a special class of short single-stranded RNA, so-called micro-RNA (miRNA), with 19–25 nucleotides, is encoded in the genomes of most multicellular organisms specifically to silence transcribed genes in order to regulate protein translation. miRNAs are synthesized in a different but similar manner to siRNA. miRNAs are assumed to silence genes partially complementary to the miRNA sequence simply by hybridizing to them. There are several other classes of short RNA in the cell, some of which have related functions and some of whose function is not known [130].

The RNAi process is especially interesting because it is not only one of the basic regulatory processes established by nature that is conserved over a wide range of taxa, but it can also be employed effectively in the laboratory in order to "reversely" knockout genes in species without a high reproductive rate [11], is less labor-intensive than classical gene knockout and facilitates controlling gene expression over time. The technology even bears a high promise of being applicable in therapeutic settings [44, 116, 123].

The computational biology community has recently taken very strongly to the topic of RNAi. Bioinformatics efforts can be grouped into (i) assembling databases of (predicted and/or validated) short RNAs and their (predicted and/or validated) targets [64,78,156,159], (ii) developing computational methods for finding miRNAs in genomes [16, 63, 157, 175], (iii) analyzing miRNAs structurally, and (iv) searching for targets for miRNAs and analyzing miRNA–target interactions [86, 109, 144, 163]. Some of the methods analyzing the sequences, secondary structures and hybridization patterns of miRNAs are addressed in Chapter 14, their role in the analysis of gene regulatory networks is addressed in Chapter 21 and RNAi is mentioned as a novel screening method in Chapter 43. A recent overview of computational methods for predicting miRNAs and their targets is given in Ref. [15].

5 Regulation – Tiling Arrays, ChIP-on-chip and Array Comparative Genomic Hybridization (Array-CGH)

Part VII of the book has concentrated on analyzing mRNA expression data. In the past few years several variants of this technology have arisen that have not been covered substantially in Part VII. The aim of this subsection is to give an overview of these techniques and address relevant bioinformatics issues if they go beyond what has been presented in Part VII.

The "traditional" view on microarrays is that they contain spots with cDNA backtransribed from mRNA sequences that have been taken from expressed genes in the cell (see Chapter 24). Thus, a microarray covers a (usually large) number of genes that can be expressed in the cell. The microarray measurement then reveals which of these genes are expressed to what levels in a given cell state. The goal of these measurements is to get a cell-wide overview of the transcription profile of all (relevant) genes. However, in order to be put on a microarray, a gene has to be known (or at least suggested) beforehand.

The so-called "tiling arrays" aim at covering not only previously known transcribed regions, but all of a genome or a substantial part of it this way [17, 152]. This affords the search for transcribed regions of the genome that have not been known beforehand. In this way, over 10 000 transcribed sequences have been found that had not been known before [17]. These findings are the starting point for finding out about the biological relevance of these previously undetected transcripts. Possible roles include previously undetected protein-coding genes, untranslated exons and RNAs with diverse functions.

A related microarray technology affords the genome-wide experimental search for transcription factor binding sites. This technology is called ChIP-on-chip – the first (ChIP) stands for "chromatin immunoprecipitation, the second for the DNA chip. The basics of the procedure are as follows [30]. Cells (ranging from yeast to mammalian cells) are grown in culture and then fixed, e.g. with formaldehyde. In this way, DNA and proteins bound to it, e.g. transcription factors, are cross-linked. DNA fragments with the cross-linked proteins attached are enriched using a procedure called immunoprecipitation. The procedure employs monoclonal antibodies binding to the proteins. Thus, the protein can be identified during enrichment. The cross-links are then reversed and the DNA is purified enriched and labeled, e.g. with a fluorescent dye. By hybridizing this DNA to a tiling microarray one can identify where specific transcription factors bind. ChIP-on-chip technology has been applied successfully to finding binding sites for individual transcription factors and mapping gene–regulator relationships cell-wide [104]. Specially configured CHiP-on-chip experiments can reveal insights into the workings of the transcription machinery and chromatin structure, where histone-modifying proteins and other chromatin players replace transcription factors. A recent overview of uses of the CHiP-on-chip technology is presented in Ref. [70].

Yet another array technology is array-CGH (CGH stands for comparative genomic hybridziation) [111]. This technique uses a tiling array in order to detect segment copy number alterations. In this case, nuclear DNA is hybridized to a tiling array with the usual readout mechanisms. Here, the resolution is a critical issue. This is the accuracy, in base pairs, to which locations of changes in copy number can be detected. This figure is a parameter of the configuration of the tiling array. Meanwhile there are tiling arrays for this purpose that cover the entire human genome down to 80-kb resolution [83,158]. Alterations in chromosomal DNA copy number are at the basis of the genomic aberrations that occur in cancer and genetically inherited diseases. Their measurement is a very effective tool for assessing type and progression state of a tumor [140]. DNA copy numbers in array-CGH data are usually more or less constant over larger regions of a genome spanning multiple DNA fragments deposited on the array (Figure 3). This is a property not usually shared by other microarray data. A characteristic bioinformatic problem in analyzing array-CGH data is therefore to detect the locations in the genome where copy numbers change. Since the data are noisy, this is a statistical problem. Several bioinformatics platforms have been developed for managing array-CGH data [34, 35, 92, 110, 118, 173]. Statistical methods for identifying such points and the intermediate copy numbers have been put forth in Refs. [90, 137, 174].

In summary, microarray technology is an effective method of measurement of the presence and abundance of biomolecules on the genomic and transcriptomic level with a wide variety of applications. The technological variants will continue to be source of new kinds of bioinformatical analysis.

6 Regulation – Epigenetics

The epigenetics level is another level at which gene regulation takes place, and which is just at the beginning of being investigated systematically with experimental and bioinformatics methods. The term epigenetics covers all stable and heritable (or potentially heritable) changes in gene expression that do not entail a change in DNA sequence [85]. Such phenomena play a significant role in development, ageing and disease.

In order for a gene to be expressed it must be accessible to the molecular transcription machinery of the cell. However, chromosomes are packed up in the chromatin in ways that are only partially understood today and are beginning to be analyzed at a genome-wide scale [112]. Different molecular mechanisms that are controlled enzymatically protect the gene from being transcribed or expose it to the transcription machinery.

We summarize these processes as they are found in mammals. The process that is closest to the DNA sequence is methylation. Methylation involves the attachment of a methyl group to the cytosine in a CpG pair (Figure 4). Methylation is catalyzed by the enzymes DNMT1, DNMT3A and DNMT3B, i.e. several methyltransferases. In the human genome, almost all interspersed CpG dinucleotides are methylated [149], which has been interpreted as a host defense mechanism against retrotransposable elements. However, so-called CpG islands which are characterized by a low degree of CpG depletion, i.e. a comparatively large amount of CpG dinucleotides, often remain unmethylated. Such islands tend to occur in the vicinity of gene promoters. A methylated CpG island protects its nearby gene from being transcribed – the gene is effectively switched off. Differential methylation of the maternal and paternal allele of a gene is the basis for the mechanism of imprinting that selectively switches off maternal or paternal genes during development.

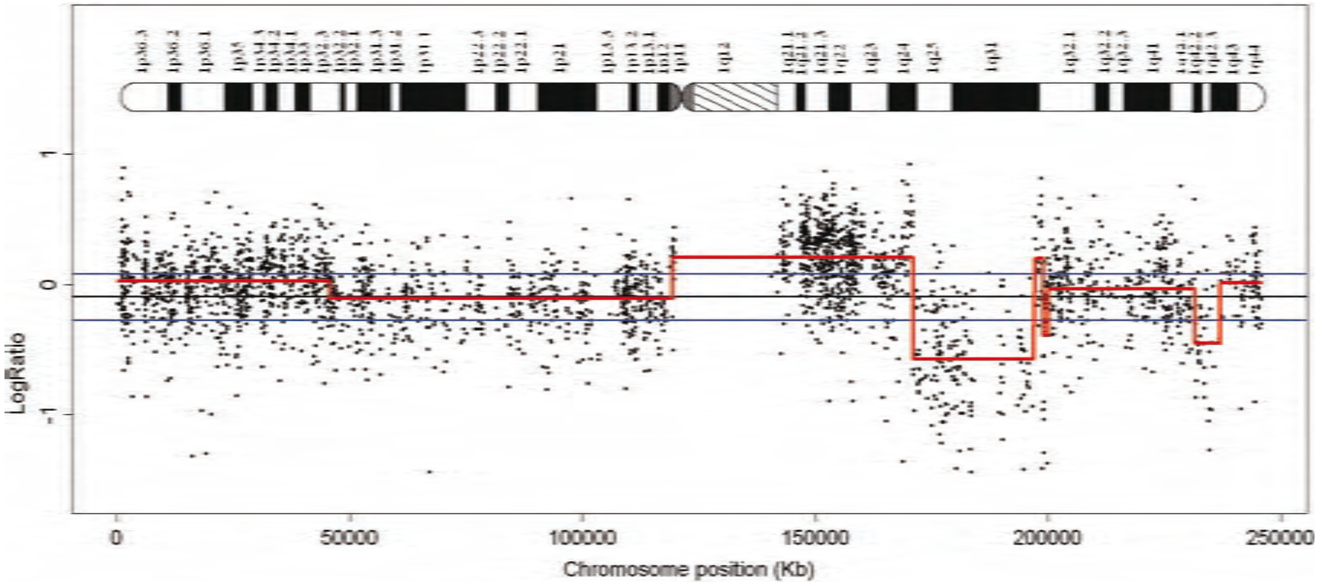


Figure 3 Profile of relative copy numbers along chromosome 1 derived from array-CGH data of a prostate cancer sample. The point cloud represents the CGH data. The baseline in black represents the average copy number; upward (downward) deviations represent larger (smaller) copy numbers. The red line is a fitted piecewise-constant function. This reflects the underlying model that the copy numbers change in steps representing losses or gains of chromosomal segments.

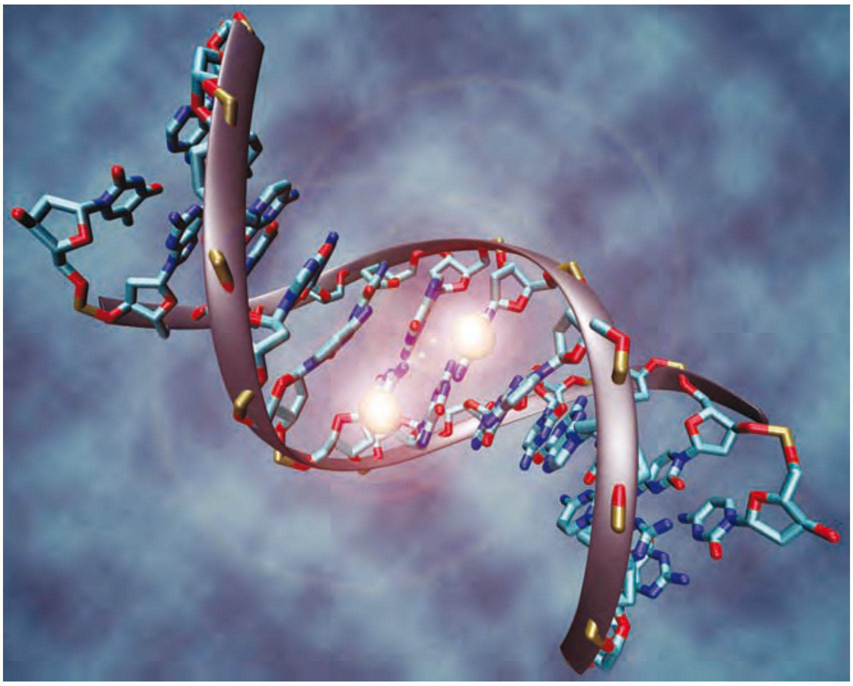


Figure 4 Rendering of a methylated CpG. The two methyl groups are enlarged for better visibility.

There are several experimental methods for measuring methylation patterns along DNA sequences with different levels of positional resolution along the DNA sequence [127]. The process of bisulphate-assisted sequencing affords single-base resolution, in principle [56, 68]. A method involving restriction enzymes has low sequence resolution and is used to generate data only at the CpG-island level rather than at higher resolution [182] – a disadvantage that is mitigated by the fact that CpG-island methylation is highly bimodal (i.e. CpG islands tend to be completely methylated or completely unmethylated). A third method uses a micorarray-based immunocapturing approach similar to the ChIP technology which achieves a resolution of about 80 kb [176] (see also Chapter 21) and is expected to achieve a resolution of about 1 kb as genome-wide high-resolution tiling arrays become available. For many questions a resolution which is less than a single base is sufficient, since methylation happens in clusters along the DNA sequence.

Methylation data are collected in several projects the most comprehensive of which is the Human Epigenome Project (http://www.epigenome.org/index.php) [141], which is carried out by an EU-funded consortium. Attempts to further coordinate epigenetic data collection are underway [89]. An important challenge to the epigenetics community will be to establish standardized data formats and quality control in a similar way as it was done within the expression microarray community ([29] and also Chapter 24). Bioinformatics can contribute to this process both by standardization, and by providing tools and databases in order to simplify adhering to the standards [4, 23].

Bioinformatics analysis of biologically or medically relevant epigenetic problems is a currently emerging field which has been named *computational epigenetics* [22]. Relevant research problems in this direction include the following. (i) Does DNA sequence influence methylation patterns? If so, which aspect of DNA sequence is the most informative one for methylation and how large is the respective correlation? Initial results indicate that DNA sequence does influence methylation substantially. Feltus and coworkers detected an influence of DNA sequence features on aberrant methylation [52]. Rollins and coworkers [149] have published a genome-wide analysis that finds relations between methylation status and CpG depletion in different cellular compartments based on data generated with the restriction enzyme method. Bock and coworkers [22] have analyzed the data by Yamada and from the Human Epigenome Project, and found that DNA sequence of a CpG island and its neighborhood is quite indicative of methylation status. The most informative aspects of sequence with correlation coefficients around 0.5 and higher include CG-rich sequence motifs, repetitive DNA and aspects of DNA 3-D structure, notably the rise and the twist of the DNA double helix [131]. (ii) Can we predict methylation state from sequence [19]? The fact that aspects of sequence are informative about methylation status suggests that we can predict methylation status from sequence [19]. In fact, Bock and coworkers have devised a method based on support vector machines (SVMs) that achieved more than 90% accuracy in both a cross-validation experiments and a blind test [22]. (iii) In general, a wide host of data-mining methods can and surely will be applied to finding patterns in methylation data and investigate their correlation with phenotype. (iv) How can we support the effective design of epigenetic cancer biomarkers with computational methods [98]?

While methylation is at the center of current experimental and bioinformatics efforts in epigenomics, there are other levels of regulation in epigenomics. Methylation only concerns a modification of the DNA molecule. In chromatin the DNA is wrapped around histone in an intricate and, today, quite well understood fashion. Making the DNA sequence accessible to the transcription machinery involves unwrapping it. This process is much less well understood, but elements of it are being uncovered. Certain post-translational modifications of histone proteins that affect their structure and thus the packing density of chromatin are a central element of this process. The pattern of post-translational modification of the different histones in the chromatin complex is called the histone code and is a major object of study in experimental epigenetics. The histone code is read by proteins with specific domains – the bromodomain and the chromodomain. These proteins initiate downstream biological responses pertaining to chromatin packing. The whole process is of great importance in diseases like cancer. Modelling these phenomena with computational methods and applying pattern recognition methods to the relevant data is still in the future. The ENCODE Project [1] is already producing a deluge of data in this area, for which efficient analysis methods have yet to be devised.

In summary, epigenetics forms the bottom-most level of what can be considered a three-level hierarchy of gene regulation. Methylation and histone modification provide the most durable control over time of which genes are expressed. This is assumed to be the level on which cell types are being defined. At the second level, transcription factor binding controls more short-term expression levels by processes that react to changes in cell state and environment. Finally, at the topmost level, RNAi secures the gene expression process against external downstream manipulations (e.g. from pathogens) and enables a more specific control than transcription factor binding (see Section 5). Ultimately, gene regulation will only be understood by discovering how these three levels interact and cooperate with mRNA degradation processes and regulatory processes at the protein level (see Section 3).

7 Protein Function – Alternative Splicing

Alternative splicing is an essential ingredient of the biology of higher eukaryotes. It is one of the two central processes by which the protein universe is expanded beyond the number of genes (the other being post-translational modification; see Section 2). Despite this, alternative splicing is only addressed at one place in this book, i.e. in Chapter 3, and there it is mainly discussed as an obstacle for accurate gene prediction. Therefore, we add a short section on the topic here.

The estimated number of human genes that can be alternatively spliced reach from 50% [128] to over 80% [120]. The average number of splice variants per gene is around 3 [181]. The average number of splicing fragments per multi-exon gene is 3.5 [181]. Genes from different functional classes tend to have different numbers of splice variants [128]. Thus, there are about 3 times as many human proteins as there are genes, bringing the estimated number of human proteins to about 80 000. (Post-translational modification adds another factor of about 10 to this, resulting in just under 1 million protein variants.) The number of splice variants can lie in the tens of thousands, such as for the *Dscam* gene in *D. melanogaster* that is responsible for neural development of the fly. Genes are spliced specifically, e.g. depending on cell type or developmental stage. Therefore, traditionally splicing has been investigated in specific, usually simple model systems with few splice variants. Splicing is effected by a macromolecular complex called the spliceosome which assembles around the (immature) pre-mRNA in order to implement the splicing operations. The assembly of the spliceosome and binding to the pre-mRNA is effected by a tissue-specific combinatorial interplay of sometimes more, sometimes fewer regulators. Similar to transcription, regulators can have a positive (agonistic, enhancer) or negative (antagonistic, silencer) effect. This results in the implementation of what might be called "cellular splicing codes". However, exactly how these codes are defined remains unclear [120].

More recently, global approaches have been followed for identifying splice variants. These approaches are traditionally based on the collection of expressed sequence tags (ESTs) [124]. Bioinformatic fragment clustering together with appropriate screening procedures in order to weed out false positives reveals splice variants. More recently, microarrays have been applied [87]. Here, alternative splice forms show up as characteristic expression profiles that, if there are only few variants, are qualitatively different and thus can usually be easily discerned from changes in gene expression. As the number of splice variants increases, the profiles are less and less easily distinguished from changes in gene expression. Bioinformatics approaches to this problem have been presented by Le and coworkers [100] and Pan and coworkers [132]. Spliceforms can also be found via comparative genomics [160]. Databases of splice variants of genes include ASD [162], ASP [181], DEDB (for *D. melanogaster*) [101], FAST DB [42], HOLLYWOOD [77] and SpliceInfo [79].

Splice variants of a gene can be visualized plausibly in a graph form (see Figure 7 in Chapter 3). There are different approaches to constructing such graphs [73, 102]. Splice forms are paths through this graph which can be enumerated completely, in principle [107]. From this collection those splice forms have to be deleted that are unlikely to be expressed. Several methods have been devised for doing so [128, 181].

Most of the splice variants seem to have functional relevance, such as the in-frame addition or removal of a functional unit. Here, the insertion or deletion of complete functional domains occurs more frequently than expected by chance, in contrast to the disruption of domains [95]. The resulting splice variants can differ in structure not only marginally, but fundamentally [165]. However, there are also inclusions or excisions of short fragments during splicing. Such variants choose between different alternatives for the N- or C-terminus, for instance, or insert or delete short internal segments that tend to have a nonloop structure and exert important influences on protein function [177].

The global investigation of splicing regulation is also underway. For instance, given a regulating protein, one can search for regulatory sites in the pre-mRNA to which it binds. *In vitro* approaches towards this goal include SELEX (systematic evolution of ligands by exponential enrichment) by which regulatory sequences are determined [31,93]. There are also *in vivo* approaches based on the transfection of cells. Most interesting for us are the computational approaches. These are usually based on a data set of sequences that are assumed to be enriched in binding sites of interest. Short motifs (e.g. hexamers) are then searched in these data sets. Significantly enriched motifs are assumed to be putative binding sites.

Conversely, given some regulatory sequence, one can search for the regulating protein. Variants of the ChIP technology discussed in Section 5 can be applied here. Combinations of the two techniques SELEX and CHiP can be used to hone in on regulators and regulatory sequences in a more focused fashion.

The hardest problem is to assign functions to regulators in terms of determining the resulting splicing pattern. This would be tantamount to resolving the cellular splicing code. However, there is compelling evidence that this code reaches beyond mRNA regulation and involves post-translational modification of regulators. Deciphering it will likely necessitate investigating the kinetics of the whole process [120].

Reviews on alternative splicing are presented in Refs. [103,120,124].

8 Interaction Networks – Immunoinformatics

Immunoinformatics [24] is the term used for bioinformatics efforts that are aimed at analyzing and modeling aspects of the immune system and its components. The field is has been forming for over half a decade – it is also called *computational immunology* [136] or *immunological bioinformatics* [113]. If applied on a cell-wide basis, one speaks of *computational immunomics* (http://research.i2r.a-star.edu.sg/IIMMS/). This book handles important aspects of immunoinformatics in Chapter 40.

Immunoinformatics aims at analyzing, modeling and predicting molecular aspects of the immune system. A major goal of immunoinformatics is the development of effective vaccines. Another is to understand the evolution of a pathogen in the presence of selective pressure exerted by the host's immune system. A third can be to analyze the effect that the diversity of the immunotypes in a population has for the spread of an epidemic.

Many people believe the immune system to be the most complex biological subnetwork in mammals and bioinformatics of the immune system can be regarded as a limited variant of general bioinformatics that comprises all levels of analysis that we have discussed in this book, from the assembly of building blocks via the analysis of binary and ternary interactions to the analysis of interaction networks and the integrated simulation of several phases spanning protein degradation, transport, molecular recognition and activation of downstream immune processes. The maturation of the field also mirrors that of all of bioinformatics – a first emphasis is on the assembly of building blocks and the analysis of molecular interactions. Interaction networks and integrated simulation follow later, and are still in the early stages of development.

This section does not afford enough space to go into detail on this subject. Two books [24, 113] are suggested for a more detailed account of the subject. However, we will touch briefly on those problems on which most research efforts are expended. Very roughly, the immune systems affords protection against foreign invaders and harmful mutants arising internally (such as malignant tumor cells). The basic molecular processes of the immune system comprise recognition of such harmful substances, specific constituent molecules called *antigens*, and mounting an immune response as the result of such recognition. Immunoinformatics today focuses on the first part – recognition. This recognition can happen directly in the extracellular fluids, such as the blood serum, by proteins termed *antibodies* (humoral immune response) or on the surface of cells after the antigen has been processed internally to the cell (cellular immune response). Cellular recognition is afforded by intracellular processes that digest relevant proteins into small peptides that are presented at the cell surface. In both cases, i.e. humoral and cellular immune responses, recognition of the antigen is manifested by a binding event of an agent of the immune system (an antibody for the humoral response and a T cell for the cellular response) to the surface-presented antigen, which is then also called an *epitope*. This binding event unleashes the cascade of immune responses.

Antigens can be recognized either in their complete form (in the case of humoral response) or via short peptides resulting from the digestion of a protein antigen internally to a cell (in the case of cellular response). The digested peptides are bound to specific molecules of the immune system belonging to the so-called major histocompatibility complex (MHC). MHC molecules are also equivocally called human leukocyte antigen (HLA) molecules. The resulting MHC–peptide complex is then translocated to the cell surface. MHC molecules are highly diverse throughout the human species. This diversity is the major measure of security for the species to survive an epidemic mounted by a new attacking pathogen – different individuals will be able to mount differing immune responses to the same pathogen. Thus, while many individuals may not survive any specific attack, the pathogen is unlikely to defeat the whole species. High HLA diversity thus also tends to correlate with high pathogen diversity [138]. The diversity of the molecules is also a major obstacle in vaccine design, and thus an important topic of bioinformatics study (see chapter 13 in Ref. [113]).

There are basically three classes of molecules that afford molecular recognition of antigens:

1. MHC class I molecules present intracellular antigens to cytotoxic T cells (Figure 5a).

2. MHC class II molecules present endocytosed antigens to T helper cells (Figure 5b).

3. Antibodies produced by B cells bind to undigested antigens during the humoral response.

The tasks of these molecules dictate their shape and influence the difficulty of the respective epitope prediction problem.

Ad 1: Endogenous antigens are processed by the MHC-I pathway which first digests an intercellular protein with the proteasome and then uses the peptide transporter TAP (transporter associated with antigen processing). TAP delivers the digested protein fragment to the MHC-I molecule within the endoplasmic reticulum, the MHC-I molecule binds to the peptide and the resulting complex leaves the endoplasmic reticulum. It is then transported though the Golgi apparatus and makes its way to the cell surface. The resulting peptides almost always comprise nine amino acid residues. Thus, the MHC-I molecule has a closed pocket that can accommodate a nonapeptide. This uniformity of the MHC-I binding sites also makes the bioinformatics problem of determining just which peptides bind to it easiest. Binding prediction is mostly performed by analyzing only sequence features of the antigen and not modeling the molecular complex structurally. A variety of machine learning methods have been applied to distinguish binders from nonbinders and to estimate binding affinity (see Ref. [54] and chapter 6 of Ref. [113]). During the final step of immunological recognition certain MHC-bound peptides (the epitopes) are identified by T cell receptors (TCRs).

The binding event of the peptide to the MHC-I molecule is only the final step in a cascade of processes that starts with the digestion of the relevant antigenic protein by the proteasome of the cell and continues with the transport of the peptide to the endoplasmic reticulum where it is loaded onto the MHC-I molecule. Both steps have a significant influence on the final epitope. The proteasome prefers certain peptide cleavage sites over others, and the TAP transporter exerts additional specificity on certain peptides. Both steps have been analyzed with sequence-based bioinformatics methods (see chapter 7 of Ref. [113]). However, in contrast to MHC binding, the data for proteasomal cleavage sites and for TAP-binding affinity are quite sparse, leading to comparatively poor prediction performance. Nevertheless,

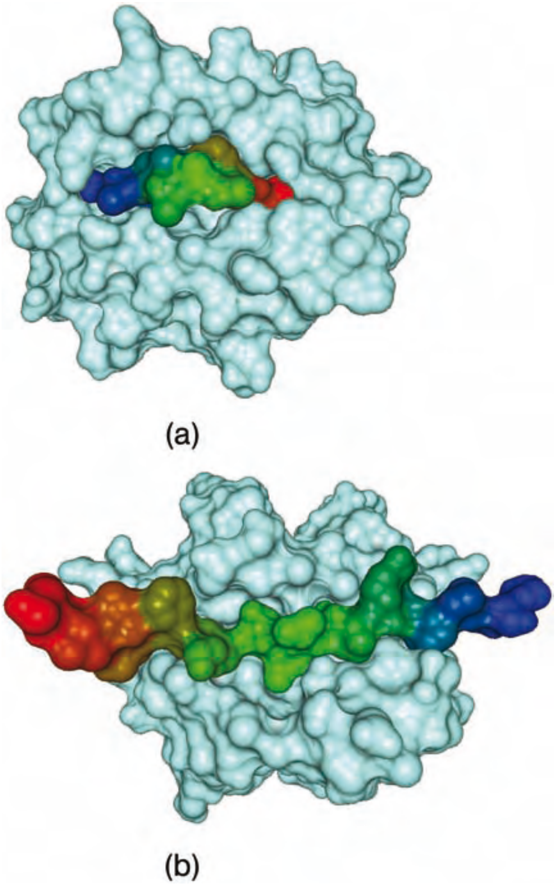


Figure 5 (a) MHC-I molecule with bound peptide: MHC-I molecules have a closed binding site and thus bind peptides with strongly restricted lengths of around 9 amino acids. (b) MHC-II molecule with bound peptide: MHC-II molecules have an open binding sites that accommodates peptides with more widely varying lengths of antibody (Images courtesy O. Kohlbacher).

solutions for an integrated analysis of all three steps of the MHC-I pathway, i.e. proteasome cleavage, TAP processing and MHC-I binding, are under development [46, 99]. See also Ref. [166] for a recent critical evaluation of the existing approaches in the light of transplantation medicine.

Ad 2: Foreign invaders are not screened from the cytosol, but from endocytosed proteins. The respective antigen-presenting pathway, i.e. the MHC-II pathway, cannot guarantee a length of nine residues for the peptide. Rather, peptides can be between eight and 20 residues in length. Thus, the pocket of the MHC-II molecule is open at both sides, antigens bind to it more diversely and the corresponding bioinformatics problem is harder. Also, integration of the analysis of the steps in the MHC-II pathway has not progressed as far as for the MHC-I pathway (see Ref. [54] and chapter 8 of Ref. [113]). This is mostly due to the lack of knowledge about the pathway.

So far, we have only discussed the analysis and prediction of surface presentation of peptides via binding of the peptide to an MHC molecule. The final recognition event is facilitated by the TCR binding to the MHC–peptide complex and eliciting an immune reaction, which qualifies the peptide as an epitope. This final step excludes about half of the MHC binders. On the basis of the respective immunological data one can use statistical learning methods such as support vector machines to predict such T cell epitopes from sequence [183, 185]. However, the TCR is extremely polymorphic, with up to billions of variants in a single individual. This probably imposes natural limits on the effectiveness and relevance of predicting binding to the TCR.

Ad 3: The humoral immune response is afforded by antibodies that are produced by B cells. B cells bind to antigens by their immunoglobulin receptors. The antibody is a soluble form of the B cell receptor (BCR) that is produced in large quantities during a humoral immune response. The BCR does not bind to digested antigen in the form of short peptides, but to the surface of undigested complex molecules. This also results in a higher structural diversity of the BCR. One of the key differences to T cell epitopes is that B cell epitopes need not be contiguous peptides. Rather, they can consist of surface patches of the antigen that are comprised of noncontiguous pieces of sequence. The bioinformatics analysis of the resulting recognition event is very hard, probably requiring the incorporation of structure, with few structure data being available Furthermore, the BCR is as polymorphic in individuals as the TCR. Thus, it is not surprising that the epitope prediction problem for the BCR is not very well understood to date ([21], see also chapter 10 of Ref. [113]).

Organism-wide data on the molecular basis of the immune systems are collectively called the immunome data [134]. Immunome data are collected in a number of databases concentrating on immune molecules [105, 147], their binding antigens [143] and both [154, 168].

The prediction methods discussed so far focus on sequence features of the epitope. Prediction methods including molecular modeling also exist, including variants of the 3-D quaternary structure–activity relationship method ([47, 186], see also Chapter 18). Structural analysis can also include docking methods [148]. Full-fledged structural analysis with molecular dynamics methods has been limited to single cases because of high computational demand, e.g. Ref. [172].

The ultimate goal behind the identification of epitopes and the analysis of their interaction with protein receptors of the immune system is the development of effective vaccines [40] (see also chapter 11 of Ref. [113]). This goal essentially amounts to selecting a peptide that is close enough to a suitable epitope from the proteome of the pathogen to be highly immunogenic, i.e. to produce the desired immune response in the vaccinated host. At the same time, the vaccine must be chosen such that a wide variety of HLA types recognize it, i.e. that it is effective in a wide part of the population [41]. The T cell epitope is critical here, because T cells are needed to orchestrate the "immune memory". Here, a single epitope may be sufficient or several selected epitopes may be necessary for eliciting immune response [126]. Not every epitope that can bind to a receptor is relevant, though. For instance, epitopes that are identical or highly similar to endogenous "self"-epitopes will not elicit an immune response. We can try to trim down the set of relevant epitopes by comparing the genomes of related virulent and avirulent pathogens, and select those for screening that are unique to the virulent strains. Comparative genomics can also help in selecting vaccine candidates from an organism that is related to, but not identical to, the pathogen – as in the case of smallpox vaccine which is taken from the cowpox virus – or one can focus on pathogen-specific proteins or those that are preferentially secreted by the pathogen in the host environment. On this selected set of antigen candidates, epitope prediction can then ensue.

Immunoinformatics is still in its infancy. The level of bioinformatics methods is largely limited to sequence analysis. The data situation is sparse. However, the field is in the process of organizing itself and is maturing more and more rapidly.

9 Cell Engineering – Synthetic Biology

Synthetic biology has been a dynamically rising paradigm in the past few years, even though the idea has been around for several decades, and corresponding research has enriched biology and biotechnology for quite some time. Synthetic biology is the result of marrying biology with engineering. Both areas have fundamentally different character and this fact is at the foundation of many of the complexities that underlie the quest of understanding living systems. Let us contrast the evolutionary approach taken by nature with the rational approach taken by an engineer. There are fundamental differences on at least two counts. (i) The (human) engineer has comparatively little resources to build a system (a few hundred or thousand person-years at most, for a single design project). In contrast, Nature develops living systems over thousands or millions of years within large populations. (ii) The engineer requires understanding of his system. This is not an issue for Nature. Nature is thought to work solely to optimize the fitness of the organism or species, respectively, in its complex environment. As a result, engineered systems tend to be modular and exhibit high degrees of symmetry. For instance, computer chips with many millions of transistors result from highly regular designs, in which at most a few thousand transistors have been inspected individually and the others arise by replication. Natural systems do display modularity, at times, but have much less of a separation between different roles and functions of their components. One example is the multiple functions observed for many proteins.

Practically throughout the whole book we have stressed the analysis approach towards understanding the function of living systems. The only place in which an engineering approach was followed was in drug design (Chapters 16, 18 and 19), which can be thought of as a variant of synthetic chemistry. Still, an engineering approach can incur substantial benefits not only for chemical, but also for biological systems. In an engineering approach to biology we would not inspect the unaltered evolved biological system, but we would change it, often with the idea to simplify it substantially. The goal of the change would be to enhance desired properties of the system or to decouple parts of it from the complex biological environment, in order to be able to study them without undesired interference by processes that are not the object of the current study. In the past several components of biological systems have been the objects of an engineering approach, such as DNA and proteins. More recently, biochemical networks were also subjected to engineering endeavors. Today, there is significant effort in engineering cellular subsystems for various purposes. This quest is still predominantly an experimental one, but it is closely linked to modeling, which can serve as a rather easily manipulable platform for early conceptualization and parameter analysis, and thus help to guide experimentation. This is why this new approach to biology is also of importance in the context of this book.

We will now briefly summarize some of the developments that have taken place in synthetic biology [14].

9.1 Genetic Engineering

Recombinant DNA can be considered as an early and very effective application of the engineering approach to genetics. The technology for manipulating DNA sequences for biotechnology purposes is far advanced. Beyond directly biotechnological purposes such as transporting genetic elements into host organisms or knocking out specific genes attempts have also been made to use DNA manipulation as a tool for performing computation (*DNA computing* [2], see also Refs. [155, 167] for more recent variants on this theme), although this quest has never matured to substantial applicability. Furthermore, engineering projects have attempted to alter the DNA backbone such as to lead to a molecule with comparable recognition properties that can more easily pass through cell membranes in order to silence DNA inside the cell or expanding the nucleobase alphabet in synthetic approaches to systems emergent properties of life [14].

9.2 Protein Engineering

Proteins also have been objects of engineering approaches. Biotechnological objectives were to create proteins that fold more stably under natural or unnatural conditions (e.g. with respect to temperature or pH [106]), or to design proteins that selectively recognize target molecules [20] or have some desired catalytic function [49]. Also, there is activity in the molecular engineering of motor proteins [169]. This field overlaps with the field of nanobiotechnology [10, 133]. In addition, many of the studies follow basic research goals of being able to understand and predict protein-like behavior like folding properties [96]. Proteins are engineered by modifying the amino acid chain of natural proteins, either in an educated trial-and-error fashion or guided by design principles such as rational design [76] or guided evolutionary design [50] supported with laboratory or software tools [49]. Building units can be either single amino acids or more complex subunits of the protein such as secondary structure elements [62]. In general protein engineering is a high complex task, because global properties of the protein cannot be expected to result from a simple combination of independent contributions of its amino acids.

9.3 Genetic Networks

The recent surge in synthetic biology came as engineers first were able to manipulate genetic networks according to preconceived design goals. One breakthrough was the design of a genetic toggle switch using interacting repressor genes in *Escherichia coli* [51, 59]. The toggle switch or flip-flop, as it is called in engineering, is a bistable element that is a basic component of many electronic systems involving memory. Logic gates such as AND, OR and an inverter can be engineered as well [72], making the vision of engineering general cellular logic approachable [67]. The interior calculation performed by a gene network could be linked to phenotypic readout, thus making the result of the calculations visible from the exterior world [94]. Furthermore, more finely graded responses could also be evoked from the engineered operation of genetic networks. One basis for this is to explore the analogous structure–function relationships at the level of protein domains [48]. In general, an entire engineering discipline is embarking on utilizing biological components [121]. The field is embarking beyond the single cell onto controlling cell–cell communication in multicellular systems. Modeling is an important aspect of synthetic biology [9]. The handling of noise is a characteristic problem of developing circuitry that has to function in an intracellular environment [161]. Understanding this phenomenon will help us greatly to advance our understanding of intracellular processes.

One can envision three goals of synthetic biology of gene circuits. The first is to study biological processes at the level of intentionally simplified circuitry in order to better decouple phenomena that are highly intertwined in evolved organisms. The resulting insights could be the basis for more adequate models. The second is to design non-natural circuits that perform as desired *in vivo*. By doing so one can investigate what are the advantages of evolved circuits over engineered ones, thus again affording an avenue towards understanding hidden secrets of biological design. Probably the most challenging question is how an engineered circuit operates in the environment of an evolved organism. The ultimate test here would be to replace an evolved circuit with an engineered one [161]. While we would not expect this to lead to better designs, we would certainly learn much about why Nature is doing things the way she does.

10 Imaging

Imaging is rapidly rising in importance in bioinformatics. Since vision takes such a large part of our sensory input, it is natural that we want to express data in images. So far, molecular structures have been the prime target of representation via (3-D) images (see Part IV of the book). The field is at the brink of expanding efforts in imaging into many different areas, spanning the cellular to the organism level. In fact, it is a plausible prognosis to make that image analysis will be a substantial aspect of bioinformatics in the years to come. Chapter 43 contains a technical discussion of some aspects of light microscopy imaging. In this section we hint at a few additional results and trends in the area of imaging biological systems.

10.1 Obtaining Pictures of Cellular Structures

There are several research efforts directed at getting more global images of cellular structures. Here, we discuss an effort to get 3-D (still) images of cells through electron tomography. In this approach, the cell is frozen and, in this state, subjected to a variant of tomography that applies electron beams instead of X-rays. Electron beams are applied in order to achieve higher resolution. Just as in traditional X-ray tomography, appropriate deconvolution procedures on the resulting data afford a 3-D picture of the cell [12]. Current technology is at a resolution of 40–50 Å. An improvement to 20 Å is considered possible. The goal is to be able to discern macromolecular complexes, such as ribosomes, proteasomes, ion channels, components of the cytoskeleton and the like. At 40-Å resolution, large complexes can be distinguished. At 20-Å resolution, we should be able to discern medium-size complexes. Image analysis then incorporates a template-based approach [25, 55], in which a database of images of the relevant macromolecular complexes is available. The templates are matched with relevant parts of the 3-D image in order to identify the locations of the respective complexes. Then, if available, the low-resolution structures can be overlaid with representative high-resolution images of the complexes affording, in principle, a high-resolution image of the complete cell. The construction of template libraries can be supported with the data from structural genomics initiatives [151] (see Chapter 13). Problems with low resolution, low signal-to-noise ratio, missing data resulting in nonisotropic resolution and crowding of the macromolecules still lie in the way of clearly separating different molecular complexes [66, 153]. Thus, intermittently, an approach mixing experimental with computational procedures employs labeling proteins with accessible epitopes using ligands carrying gold nanoparticles. Such proteins can then be identified more easily in the macromolecular crowd. Labeling works better for membrane proteins with extracellular epitopes than inside the cell, where it requires non-invasive genetic manipulations.

Images of cells from *Dictyostelium discoideum* [122] and *Spiroplasma melliferum* [97] have been generated in this way, and afforded unprecedented insights into the internal organization of the cell (Figure 6). In general, it is impressive how much our intuitive picture of a cell is changed by the availability of global cell images. The cell is an extremely crowded environment – thicker than lentil soup – and notions like transport and diffusion receive a different connotation in the presence of such images.

The EU is funding a Network of Excellence on 3-D electron microscopy that bundles many of the European activities in the area
(http://www.3dem-noe.org).

There are also cell imaging efforts based on light microscopy [164]. With the advent of green fluorescent protein (GFP) technology which can now utilize several colors, fluorescence labeling in order to visualize intracellular molecules, structures and processes has received a vital push. GFP is used to visualize targeted structures in cells, such as G-protein activity, special signaling events and calcium dynamics. Confocal light microscopy, which only collects light emerging from the focal plane, yields 3-D images, but with live samples the classical wide-field microscopy with subsequent computational deconvolution of the data is still the preferable technology for this purpose.

In addition, special laser technology affords us with the facility to press the resolution of light microscopy below the usual limit of about half the wavelength of the applied light [74,75]. This technology, which is just emerging, has

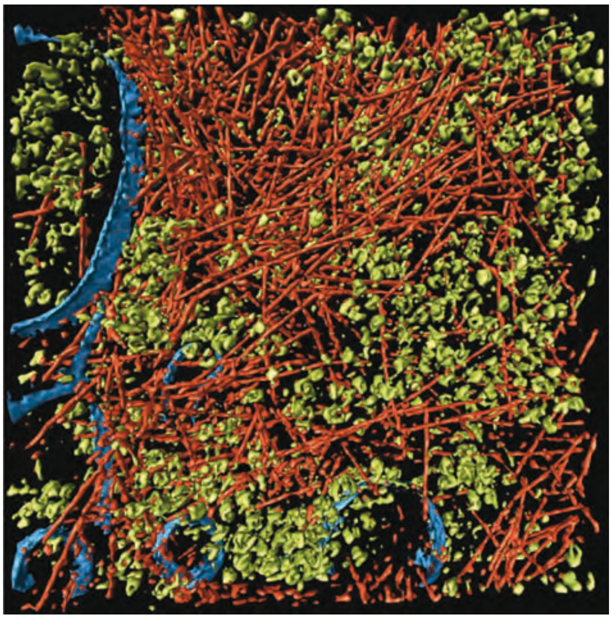


Figure 6 Visualization of part of a cell with intracellular structures. Actin filaments (reddish), other macromolecular complexes, mostly ribosomes (green), and membranes (blue). (From Ref. [122].)

the advantage that, in principle, it is applicable to live samples rather than just fixated cells. Making this technology work for biological specimens would be no less than a breakthrough.

Chapter 43 contains a more detailed discussion of light microscopy imaging of cellular structures and processes.

10.2 Movies of Cellular Processes

So far, we have discussed still imaging of cellular structures. Of course, getting direct information along the time dimension is another very effective input into the analysis of biological processes. Thus, 4-D imaging of cellular structure, i.e. making movies of them, is a vivid research field [61]. The restraint that the sample be viable throughout the experiment is mandatory here, of course, and can be a great challenge. The other significant concern is the limited amount of light output that is usually available in such an experiment.

An especially attractive target for 4-D imaging has been the mechanics of eukaryotic cell division. Movies have been made on how chromosomes rearrange during the early stages of mitosis [61] and of the rearrangement of the mitotic spindle during the later stages of mitosis [65]. The cytoskeleton is an especially interesting target because of its relative rigidity [13]. Other cellular structures are more amorphous, making quantitative structural measurements very difficult. In order to deal with this problem, artificial landmarks can be introduced in cells in an approach called pattern photobleaching. Virtual reality viewers are employed to visualize the resulting data [60]. The resulting data can also be input to quantitative models of the investigated cellular processes [65].

10.3 Organism Development

Imaging can go beyond single cells and cover the whole organism. One special case is monitoring the location of prespecified expressed [184] genes, the movement of cells [146,180] or the development of morphology of cellular structures [178] during organism development. Such studies are being undertaken, e.g. in *Drosophila* and *Xenopus* [170], with a certain focus on neuronal development. The analysis of data resulting from *in situ* imaging of expressed genes affords special bioinformatics techniques that merge image analysis (Chapter 43) with the analysis of expression data (Chapters 24ff.). First steps in this direction have been undertaken [135].

11 Modeling Organs

From a basic research point of view, the ultimate vision of computational biology could be formulated as the ability to simulate biological processes on a hierarchy of levels, starting from the bottom-most level that considers molecular processes, and reaching to higher levels that, ultimately, represent the macroscopic physiology and biomechanics of organ systems. The notion of the *physiome* has been coined to encompass the totality of all physiologically relevant information pertaining to this modeling hierarchy. While such a comprehensive goal may seem utopic, there are actually concerted efforts embarking along this path. The International Union of Physiological Sciences has introduced the Physiome Project [38, 80, 81]. This project provides a web site (http://www.physiome.org/) that collects world-wide contributions towards filling in the gigantic simulation landscape presented by the goals of the project. At heart, the Physiome Project is a computational project. However, completing it requires a wide variety and large volume of new data. In order to reach through the different model hierarchies, effective approaches towards multiscale modeling are of central importance [39]. The Physiome Project is structured into model subcategories. In the past, physiological modeling has concentrated on a few organ systems. The heart is definitely one of the best researched organs, in this respect, with detailed models being available from the electrophysiology of the heart at the molecular level over the microstructure of cardiac tissue and the organization of fibers in the heart muscle to mechanical models for heart beat that include the spread of electrical activation wavefronts throughout the heart muscle and the pressures in the coronary arteries [82, 129]. These models have to be linked together now. Blood flow [32, 33] through the lung has also been modeled at an organ-wide level. A proposal has been made to model bone [43].

Of course, physiome modeling has direct medical application, since disease is an aberration of normal biology and can probably best be understood as such. However, there are also modeling efforts that are directly targeted at disease, i.e. the direct modeling of tumor growth. Tumor growth is the result of several processes such as rapid cell proliferation with loss of apoptosis, tissue invasion and angiogenesis. Lastly, the effect of radiation therapy and chemotherapy can be modeled. Models for many of these aspects have been developed independently, some of which are already quite interdisciplinary (e.g. see Ref. [36] for cell proliferation, Ref. [139] for tissue invasion, Refs. [8, 108] for angiogenesis, Refs. [58, 145] for the effect of drug treatment and Ref. [45] for radiation therapy). The adaptation of such models to each other and their integration is still for the future, but people are working in this direction [3].

12 Outlook

This points addressed in this chapter are by no means comprehensive. Rather, they reflect the exposure and judgment of the author with respect to emerging and promising themes. Many of the research activities in these areas are currently still driven by experimental issues, but all of them harbor great potential for further bioinformatic development. Probably it would be a futile exercise to attempt a comprehensive balanced overview of all promising open leads in bioinformatics and the experimental technologies relevant for the field, anyway. However, as a collection, these topics bear witness to the tremendous dynamics of the field. Bioinformatics is going to continue for quite some time to be a productive source of extremely exciting scientific problems that combine great challenges for basic research with often immediate application relevance. In the view of the author, these characteristics presently single out bioinformatics among all scientific disciplines.

Acknowledgments

I thank Mario Albrecht, Iris Antes, Christoph Bock, Francisco Domingues, Oliver Kohlbacher, Claus-Wilhelm von der Lieth, Jochen Maydt, Jörg Rahnenführer, Kirsten Roomp, Laura Tolosi, Martin Vingron and Nils Weinhold for helpful suggestions on this chapter, pointers to relevant literature and the preparation of figures.

References

1 ENCODE Project Consortium. 2004. The ENCODE (ENCyclopedia Of DNA Elements) Project. Science **306**: 636–40.

2 Adleman, L. M. 1994. Molecular computation of solutions to combinatorial problems. Science **266**: 1021–4.

3 Agur, Z., L. Arakelyan, G. Belilty, et al. 2004. Application of the virtual cancer patient engine (VCPE) for improving oncological treatment design. J. Clin. Oncol. **22**: 692.

4 Amoreira, C., W. Hindermann and C. Grunau. 2003. An improved version of the DNA Methylation database (MethDB). Nucleic Acids Res. **31**: 75–7.

5 Aoki, K. F., H. Mamitsuka, T. Akutsu and M. Kanehisa. 2005. A score matrix to reveal the hidden links in glycans. Bioinformatics **21**: 1457–63.

6 Aoki, K. F., N. Ueda, A. Yamaguchi, M. Kanehisa, T. Akutsu and H. Mamitsuka. 2004. Application of a new probabilistic model for recognizing complex patterns in glycans. Bioinformatics **20 (Suppl. 1)**: i6–14.

7 Aoki, K. F., A. Yamaguchi, N. Ueda, T. Akutsu, H. Mamitsuka, S. Goto and M. Kanehisa. 2004. KCaM (KEGG Carbohydrate Matcher): a software tool for analyzing the structures of carbohydrate sugar chains. Nucleic Acids Res. **32**: W267–72.

8 Arakelyan, L., V. Vainstein and Z. Agur. 2002. A computer algorithm describing the process of vessel formation and maturation, and its use for predicting the effects of anti-angiogenic and anti-maturation therapy on vascular tumor growth. Angiogenesis **5**: 203–14.

9 Arkin, A. P. 2001. Synthetic cell biology. Curr. Opin. Biotechnol. **12**: 638–44.

10 Astier, Y., H. Bayley and S. Howorka. 2005. Protein components for nanodevices. Curr. Opin. Chem. Biol. **9**: 576–84.

11 Baum, B. and G. Craig. 2004. RNAi in a postmodern, postgenomic era. Oncogene **23**: 8336–9.

12 Baumeister, W. 2005. From proteomic inventory to architecture. FEBS Lett. **579**: 933.

13 Bement, W. M., A. M. Sokac and C. A. Mandato. 2003. Four-dimensional imaging of cytoskeletal dynamics in *Xenopus* oocytes and eggs. Differentiation **71**: 518–27.

14 Benner, S. A. and A. M. Sismour. 2005. Synthetic biology. Nat. Rev. Genet. **6**: 533–43.

15 Bentwich, I. 2005. Prediction and validation of microRNAs and their targets. FEBS Lett. **579**: 5904–10.

16 Bentwich, I., A. Avniel, Y. Karov, et al. 2005. Identification of hundreds of conserved and nonconserved human microRNAs. Nat. Genet. **37**: 766–70.

17 Bertone, P., V. Stolc, T. E. Royce, et al. 2004. Global identification of human transcribed sequences with genome tiling arrays. Science **306**: 2242–6.

18 Beyer, A., J. Hollunder, H. P. Nasheuer and T. Wilhelm. 2004. Post-transcriptional expression regulation in the yeast *Saccharomyces cerevisiae* on a genomic scale. Mol. Cell. Proteomics **3**: 1083–92.

19 Bhasin, M., H. Zhang, E. L. Reinherz and P. A. Reche. 2005. Prediction of methylated CpGs in DNA sequences using a support vector machine. FEBS Lett. **579**: 4302–8.

20 Binz, H. K., P. Amstutz and A. Pluckthun. 2005. Engineering novel binding proteins from nonimmunoglobulin domains. Nat. Biotechnol. **23**: 1257–68.

21 Blythe, M. J. and D. R. Flower. 2005. Benchmarking B cell epitope prediction: underperformance of existing methods. Protein Sci. **14**: 246–8.

22 Bock, C., M. Paulsen, S. Tierling, T. Mikeska, T. Lengauer and J. E. Walter. 2006. CpG island methylation in human lymphocytes is highly correlated with DNA sequence patterns, repeat frequencies and predicted DNA structure. PLoS Genet. **2**: e26.

23 Bock, C., S. Reither, T. Mikeska, M. Paulsen, J. Walter and T. Lengauer. 2005. BiQ Analyzer: visualization and quality control for DNA methylation data from bisulfite sequencing. Bioinformatics **21**: 4067–8.

24 Bock, G. and J. Goode. 2003. *Immunoinformatics: Bioinformatic Strategies for Better Understanding Of Immune Function*. Wiley, Chichester.

25 Bohm, J., A. S. Frangakis, R. Hegerl, S. Nickell, D. Typke and W. Baumeister. 2000. Toward detecting and identifying macromolecules in a cellular context: template matching applied to electron tomograms. Proc. Natl Acad. Sci. USA **97**: 14245–50.

26 Bohne-Lang, A. and C. W. von der Lieth. 2005. GlyProt: *in silico* glycosylation of proteins. Nucleic Acids Res. **33**: W214–9.

27 Bohne, A., E. Lang and C. W. von der Lieth. 1999. SWEET – WWW-based rapid 3D construction of oligo- and polysaccharides. Bioinformatics **15**: 767–8.

28 Borman, S. and C. Washington. 2005. Carbohydrate advances. Chem. Eng. News **83**: 41–50.

29 Brazma, A., P. Hingamp, J. Quackenbush et al. 2001. Minimum Information About a Microarray Experiment (MIAME) – toward standards for microarray data. Nat. Genet. **29**: 365–71.

30 Buck, M. J. and J. D. Lieb. 2004. ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. Genomics **83**: 349–60.

31 Buratti, E. and F. E. Baralle. 2005. Another step forward for SELEXive splicing. Trends Mol. Med. **11**: 5.

32 Burrowes, K. S., P. J. Hunter and M. H. Tawhai. 2005. Investigation of the relative effects of vascular branching structure and gravity on pulmonary arterial blood flow heterogeneity via an image-based computational model. Acad. Radiol. **12**: 1464–74.

33 Burrowes, K. S., M. H. Tawhai and P. J. Hunter. 2004. Modeling RBC and neutrophil distribution through an anatomically based pulmonary capillary network. Ann. Biomed. Eng. **32**: 585–95.

34 Chen, W., F. Erdogan, H. H. Ropers, S. Lenzner and R. Ullmann. 2005. CGHPRO – a comprehensive data analysis tool for array CGH. BMC Bioinformatics **6**: 85.

35 Chi, B., R. J. DeLeeuw, B. P. Coe, C. MacAulay and W. L. Lam. 2004. SeeGH – a software tool for visualization of whole genome array comparative genomic hybridization data. BMC Bioinformatics **5**: 13.

36 Chignola, R. and E. Milotti. 2005. A phenomenological approach to the simulation of metabolism and proliferation dynamics of large tumour cell populations. Phys. Biol. **2**: 8–22.

37 Ciechanover, A. and P. Brundin. 2003. The ubiquitin proteasome system in neurodegenerative diseases: sometimes the chicken, sometimes the egg. Neuron **40**: 427–46.

38 Crampin, E. J., M. Halstead, P. Hunter, P. Nielsen, D. Noble, N. Smith and M. Tawhai. 2004. Computational physiology and the Physiome Project. Exp. Physiol. **89**: 1–26.

39 Crampin, E. J., N. P. Smith and P. J. Hunter. 2004. Multi-scale modelling and

the IUPS physiome project. J. Mol. Histol. **35**: 707–14.

40 DE GROOT, A. S. AND J. A. BERZOFSKY. 2004. From genome to vaccine – new immunoinformatics tools for vaccine design. Methods **34**: 425–8.

41 DE GROOT, A. S., E. A. BISHOP, B. KHAN, et al. 2004. Engineering immunogenic consensus T helper epitopes for a cross-clade HIV vaccine. Methods **34**: 476–87.

42 DE LA GRANGE, P., M. DUTERTRE, N. MARTIN AND D. AUBOEUF. 2005. FAST DB: a website resource for the study of the expression regulation of human gene products. Nucleic Acids Res. **33**: 4276–4284.

43 DEFRANOUX, N. A., C. L. STOKES, D. L. YOUNG AND A. J. KAHN. 2005. *In silico* modeling and simulation of bone biology: a proposal. J. Bone Miner. Res. **20**: 1079–84.

44 DENOVAN-WRIGHT, E. M. AND B. L. DAVIDSON. 2005. RNAi: a potential therapy for the dominantly inherited nucleotide repeat diseases. Gene Ther. **13**: 525–31.

45 DIONYSIOU, D. D., G. S. STAMATAKOS, N. K. UZUNOGLU AND K. S. NIKITA. 2006. A computer simulation of *in vivo* tumour growth and response to radiotherapy: new algorithms and parametric results. Comput. Biol. Med. **36**: 448–64.

46 DÖNNES, P. AND O. KOHLBACHER. 2005. Integrated modeling of the major events in the MHC class I antigen processing pathway. Protein Sci. **14**: 2132–40.

47 DOYTCHINOVA, I. A., P. GUAN AND D. R. FLOWER. 2004. Quantitative structure–activity relationships and the prediction of MHC supermotifs. Methods **34**: 444–53.

48 DUEBER, J. E., B. J. YEH, K. CHAK AND W. A. LIM. 2003. Reprogramming control of an allosteric signaling switch through modular recombination. Science **301**: 1904–8.

49 DWYER, M. A., L. L. LOOGER AND H. W. HELLINGA. 2004. Computational design of a biologically active enzyme. Science **304**: 1967–71.

50 EIJSINK, V. G., S. GASEIDNES, T. V. BORCHERT AND B. VAN DEN BURG. 2005. Directed evolution of enzyme stability. Biomol. Eng. **22**: 21–30.

51 ELOWITZ, M. B. AND S. LEIBLER. 2000. A synthetic oscillatory network of transcriptional regulators. Nature **403**: 335–8.

52 FELTUS, F. A., E. K. LEE, J. F. COSTELLO, C. PLASS AND P. M. VERTINO. 2003. Predicting aberrant CpG island methylation. Proc. Natl Acad. Sci. USA **100**: 12253–8.

53 FIRE, A., S. XU, M. K. MONTGOMERY, S. A. KOSTAS, S. E. DRIVER AND C. C. MELLO. 1998. Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegan*s. Nature **391**: 806–11.

54 FLOWER, D. R. 2003. Towards *in silico* prediction of immunogenic epitopes. Trends Immunol. **24**: 667–74.

55 FRANGAKIS, A. S., J. BOHM, F. FORSTER, S. NICKELL, D. NICASTRO, D. TYPKE, R. HEGERL AND W. BAUMEISTER. 2002. Identification of macromolecular complexes in cryoelectron tomograms of phantom cells. Proc. Natl Acad. Sci. USA **99**: 14153–8.

56 FROMMER, M., L. E. MCDONALD, D. S. MILLAR, C. M. COLLIS, F. WATT, G. W. GRIGG, P. L. MOLLOY AND C. L. PAUL. 1992. A genomic sequencing protocol that yields a positive display of 5-methylcytosine residues in individual DNA strands. Proc. Natl Acad. Sci. USA **89**: 1827–31.

57 GARAVELLI, J. S. 2004. The RESID Database of Protein Modifications as a resource and annotation tool. Proteomics **4**: 1527–33.

58 GARDNER, S. N. 2002. Modeling multi-drug chemotherapy: tailoring treatment to individuals. J. Theor. Biol. **214**: 181–207.

59 GARDNER, T. S., C. R. CANTOR AND J. J. COLLINS. 2000. Construction of a genetic toggle switch in *Escherichia coli*. Nature **403**: 339–42.

60 GERLICH, D., J. BEAUDOUIN, M. GEBHARD, J. ELLENBERG AND R. EILS. 2001. Four-dimensional imaging and quantitative reconstruction to analyse complex spatiotemporal processes in live cells. Nat. Cell Biol. **3**: 852–5.

61 Gerlich, D. and J. Ellenberg. 2003. 4D imaging to assay complex dynamics in live specimens. Nat. Cell Biol. **Suppl.**: S14–9.

62 Ghirlanda, G., J. D. Lear, N. L. Ogihara, D. Eisenberg and W. F. DeGrado. 2002. A hierarchic approach to the design of hexameric helical barrels. J. Mol. Biol. **319**: 243–53.

63 Grad, Y., J. Aach, G. D. Hayes, B. J. Reinhart, G. M. Church, G. Ruvkun and J. Kim. 2003. Computational and experimental identification of *C. elegans* microRNAs. Mol. Cell **11**: 1253–63.

64 Griffiths-Jones, S., R. J. Grocock, S. van Dongen, A. Bateman and A. J. Enright. 2006. miRBase: microRNA sequences, targets and gene nomenclature. Nucleic Acids Res. **34**: D140–4.

65 Grill, S. W., J. Howard, E. Schaffer, E. H. Stelzer and A. A. Hyman. 2003. The distribution of active force generators controls mitotic spindle position. Science **301**: 518–21.

66 Grünewald, K., O. Medalia, A. Gross, A. C. Steven and W. Baumeister. 2003. Prospects of electron cryotomography to visualize macromolecular complexes inside cellular compartments: implications of crowding. Biophys. Chem. **100**: 577–91.

67 Guet, C. C., M. B. Elowitz, W. Hsing and S. Leibler. 2002. Combinatorial synthesis of genetic networks. Science **296**: 1466–70.

68 Hajkova, P., O. el-Maarri, S. Engemann, J. Oswald, A. Olek and J. Walter. 2002. DNA-methylation analysis by the bisulfite-assisted genomic sequencing method. Methods Mol. Biol. **200**: 143–54.

69 Han, K. K. and A. Martinage. 1992. Post-translational chemical modification(s) of proteins. Int. J. Biochem. **24**: 19–28.

70 Hanlon, S. E. and J. D. Lieb. 2004. Progress and challenges in profiling the dynamics of chromatin and transcription factor binding with DNA microarrays. Curr. Opin. Genet. Dev. **14**: 697–705.

71 Hashimoto, K., S. Goto, S. Kawano, K. F. Aoki-Kinoshita, N. Ueda, M. Hamajima, T. Kawasaki and M. Kanehisa. 2006. KEGG as a glycome informatics resource. Glycobiology **16**: 63R–70R.

72 Hasty, J., D. McMillen and J. J. Collins. 2002. Engineered gene circuits. Nature **420**: 224–30.

73 Heber, S., M. Alekseyev, S.-H. Sze, H. Tang and P. A. Pevzner. 2002. Splicing graphs and EST assembly problem. Bioinformatics **18**: S181–8.

74 Hell, S. W. 2003. Toward fluorescence nanoscopy. Nat. Biotechnol. **21**: 1347–55.

75 Hell, S. W., M. Dyba and S. Jakobs. 2004. Concepts for nanoscale resolution in fluorescence microscopy. Curr. Opin. Neurobiol. **14**: 599–609.

76 Hellinga, H. W. 1997. Rational protein design: combining theory and experiment. Proc. Natl Acad. Sci. USA **94**: 10015–7.

77 Holste, D., G. Huo, V. Tung and C. B. Burge. 2006. HOLLYWOOD: a comparative relational database of alternative splicing. Nucleic Acids Res. **34**: D56–62.

78 Hsu, P. W., H. D. Huang, S. D. Hsu, et al. 2006. miRNAMap: genomic maps of microRNA genes and their target genes in mammalian genomes. Nucleic Acids Res. **34**: D135–9.

79 Huang, H.-D., J.-T. Horng, F.-M. Lin, Y.-C. Chang and C.-C. Huang. 2005. SpliceInfo: an information repository for mRNA alternative splicing in human genome. Nucleic Acids Res. **33**: D80–85.

80 Hunter, P. J. 2004. The IUPS Physiome Project: a framework for computational physiology. Prog. Biophys. Mol. Biol. **85**: 551–69.

81 Hunter, P. J. and T. K. Borg. 2003. Integration from proteins to organs: the Physiome Project. Nat. Rev. Mol. Cell Biol. **4**: 237–43.

82 Hunter, P. J., A. J. Pullan and B. H. Smaill. 2003. Modeling total heart function. Annu. Rev. Biomed. Eng. **5**: 147–77.

83 Ishkanian, A. S., C. A. Malloff, S. K. Watson, et al. 2004. A tiling resolution

DNA microarray with complete coverage of the human genome. Nat. Genet. **36**: 299–303.

84 Jensen, L. J., R. Gupta, N. Blom, et al. 2002. Prediction of human protein function from post-translational modifications and localization features. J. Mol. Biol. **319**: 1257–65.

85 Jiang, Y. H., J. Bressler and A. L. Beaudet. 2004. Epigenetics and human disease. Annu. Rev. Genomics Hum Genet. **5**: 479–510.

86 John, B., A. J. Enright, A. Aravin, T. Tuschl, C. Sander and D. S. Marks. 2004. Human microRNA targets. PLoS Biol. **2**: e363.

87 Johnson, J. M., J. Castle, P. Garrett-Engele, et al. 2003. Genome-wide survey of human alternative pre-mRNA splicing with exon junction microarrays. Science **302**: 2141–2144.

88 Johnson, S. A. and T. Hunter. 2005. Kinomics: methods for deciphering the kinome. Nat. Methods **2**: 17–25.

89 Jones, P. A. and R. Martienssen. 2005. A blueprint for a Human Epigenome Project: the AACR Human Epigenome Workshop. Cancer Res. **65**: 11241–6.

90 Jong, K., E. Marchiori, G. Meijer, A. V. Vaart and B. Ylstra. 2004. Breakpoint identification and smoothing of array comparative genomic hybridization data. Bioinformatics **20**: 3636–7.

91 Kim, P. M. and B. Tidor. 2003. Limitations of quantitative gene regulation models: a case study. Genome Res. **13**: 2391–5.

92 Kim, S. Y., S. W. Nam, S. H. Lee, W. S. Park, N. J. Yoo, J. Y. Lee and Y. J. Chung. 2005. ArrayCyGHt: a web application for analysis and visualization of array-CGH data. Bioinformatics **21**: 2554–5.

93 Klug, S. J. and M. Famulok. 1994. All you wanted to know about SELEX. Mol. Biol. Rep. **20**: 97.

94 Kobayashi, H., M. Kaern, M. Araki, K. Chung, T. S. Gardner, C. R. Cantor and J. J. Collins. 2004. Programmable cells: interfacing natural and engineered gene networks. Proc. Natl Acad. Sci. USA **101**: 8414–9.

95 Kriventseva, E. V., I. Koch, R. Apweiler, M. Vingron, P. Bork, M. S. Gelfand and S. Sunyaev. 2003. Increase of functional diversity by alternative splicing. Trends Genet. **19**: 124–8.

96 Kuhlman, B., G. Dantas, G. C. Ireton, G. Varani, B. L. Stoddard and D. Baker. 2003. Design of a novel globular protein fold with atomic-level accuracy. Science **302**: 1364–8.

97 Kürner, J., A. S. Frangakis and W. Baumeister. 2005. Cryo-electron tomography reveals the cytoskeletal structure of *Spiroplasma melliferum*. Science **307**: 436–8.

98 Laird, P. W. 2003. The power and the promise of DNA methylation markers. Nat. Rev. Cancer **3**: 253–66.

99 Larsen, M. V., C. Lundegaard, K. Lamberth, S. Buus, S. Brunak, O. Lund and M. Nielsen. 2005. An integrative approach to CTL epitope prediction: a combined algorithm integrating MHC class I binding, TAP transport efficiency, and proteasomal cleavage predictions. Eur J. Immunol. **35**: 2295–303.

100 Le, K., K. Mitsouras, M. Roy, Q. Wang, Q. Xu, S. F. Nelson and C. Lee. 2004. Detecting tissue-specific regulation of alternative splicing as a qualitative change in microarray data. Nucleic Acids Res. **32**: e180.

101 Lee, B. T. K., T. W. Tan and S. Ranganathan. 2004. DEDB: a database of *Drosophila melanogaster* exons in splicing graph form. BMC Bioinformatics **5**: 189.

102 Lee, C., C. Grasso and M. F. Sharlow. 2002. Multiple sequence alignment using partial order graphs. Bioinformatics **18**: 452–464.

103 Lee, C. and Q. Wang. 2005. Bioinformatics analysis of alternative splicing. Brief Bioinform. **6**: 23–33.

104 Lee, T. I., N. J. Rinaldi, F. Robert, et al. 2002. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. Science **298**: 799–804.

105 Lefranc, M. P., V. Giudicelli, Q. Kaas, et al. 2005. IMGT, the international

ImMunoGeneTics information system. Nucleic Acids Res. **33**: D593–7.

106 Lehmann, M. and M. Wyss. 2001. Engineering proteins for thermostability: the use of sequence alignments versus rational design and directed evolution. Curr. Opin. Biotechnol. **12**: 371–5.

107 Leipzig, J., P. Pevzner and S. Heber. 2004. The Alternative Splicing Gallery (ASG): bridging the gap between genome and transcriptome. Nucleic Acids Res. **32**: 3977–983.

108 Levine, H. A., S. Pamuk, B. D. Sleeman and M. Nilsen-Hamilton. 2001. Mathematical modeling of capillary formation and development in tumor angiogenesis: penetration into the stroma. Bull. Math. Biol. **63**: 801–63.

109 Lewis, B. P., C. B. Burge and D. P. Bartel. 2005. Conserved seed pairing, often flanked by adenosines, indicates that thousands of human genes are microRNA targets. Cell **120**: 15–20.

110 Lingjaerde, O. C., L. O. Baumbusch, K. Liestol, I. K. Glad and A. L. Borresen-Dale. 2005. CGH-Explorer: a program for analysis of array-CGH data. Bioinformatics **21**: 821–2.

111 Lockwood, W. W., R. Chari, B. Chi and W. L. Lam. 2006. Recent advances in array comparative genomic hybridization technologies and their applications in human genetics. Eur J. Hum. Genet. **14**: 139–48.

112 Loden, M. and B. van Steensel. 2005. Whole-genome views of chromatin structure. Chromosome Res. **13**: 289–98.

113 Lund, O., M. Nielsen, C. Lundegaard, C. Kesmir and S. Brunak. 2005. *Immunological Bioinformatics*. MIT Press, Cambridge, MA.

114 Lutteke, T., A. Bohne-Lang, A. Loss, T. Goetz, M. Frank and C. W. von der Lieth. 2006. GLYCOSCIENCES.de: an Internet portal to support glycomics and glycobiology research. Glycobiology **16**: 71R–81R.

115 Lutteke, T., M. Frank and C. W. von der Lieth. 2004. Data mining the protein data bank: automatic detection and assignment of carbohydrate structures. Carbohydr. Res. **339**: 1015–20.

116 Manfredsson, F. P., A. S. Lewin and R. J. Mandel. 2006. RNA knockdown as a potential therapeutic strategy in Parkinson's disease. Gene Ther. **13**: 517–24.

117 Mani, A. and E. P. Gelmann. 2005. The ubiquitin–proteasome pathway and its role in cancer. J. Clin. Oncol. **23**: 4776–89.

118 Margolin, A. A., J. Greshock, T. L. Naylor, et al. 2005. CGHAnalyzer: a stand-alone software package for cancer genome analysis using array-based DNA copy number data. Bioinformatics **21**: 3308–11.

119 Marx, J. 2005. P-bodies mark the spot for controlling protein production. Science **310**: 764–5.

120 Matlin, A. J., F. Clark and C. W. J. Smith. 2005. Understanding alternative splicing: twoards a cellular code. Nat. Rev. Mol. Cell. Biol. **6**: 386.

121 McDaniel, R. and R. Weiss. 2005. Advances in synthetic biology: on the path from prototypes to applications. Curr. Opin. Biotechnol. **16**: 476–83.

122 Medalia, O., I. Weber, A. S. Frangakis, D. Nicastro, G. Gerisch and W. Baumeister. 2002. Macromolecular architecture in eukaryotic cells visualized by cryoelectron tomography. Science **298**: 1209–13.

123 Mocellin, S., R. Costa and D. Nitti. 2006. RNA interference: ready to silence cancer? J. Mol. Med. **84**: 4–15.

124 Modrek, B. and C. Lee. 2002. A genomic view of alternative splicing. Nat. Genet. **30**: 13.

125 Moore, M. J. 2005. From birth to death: the complex lives of eukaryotic mRNAs. Science **309**: 1514–8.

126 Moreau, V., C. Granier, S. Villard, D. Laune and F. Molina. 2006. Discontinuous epitope prediction based on mimotope analysis. Bioinformatics **22**: 1088–95.

127 Murrell, A., V. K. Rakyan and S. Beck. 2005. From genome to epigenome. Hum Mol. Genet. **14 (Spec. 1)**: R3–10.

128 Neverov, A. D., I. I. Artamonova, R. N. Nurtdinov, D. Frishman, M. S.

GELFAND AND A. A. MIRONOV. 2005. Alternative splicing and protein function. BMC Bioinformatics **6**: 266.

129 NOBLE, D. 2004. Modeling the heart. Physiology (Bethesda) **19**: 191–7.

130 NOVINA, C. D. AND P. A. SHARP. 2004. The RNAi revolution. Nature **430**: 161–4.

131 OLSON, W. K., M. BANSAL, S. K. BURLEY, et al. 2001. A standard reference frame for the description of nucleic acid base-pair geometry. J. Mol. Biol. **313**: 229–37.

132 PAN, Q., O. SHAI, C. MISQUITTA, et al. 2004. Revealing global regulatory features of mammalian alternative splicing using a quantitative microarray platform. Mol. Cell **16**: 929.

133 PATOLSKY, F., Y. WEIZMANN AND I. WILLNER. 2004. Actin-based metallic nanowires as bio-nanotransporters. Nat. Mater **3**: 692–5.

134 PEDERSON, T. 1999. The immunome. Mol. Immunol. **36**: 1127–8.

135 PENG, H. AND E. W. MYERS. 2004. Comparing *in situ* mRNA expression patterns of *Drosophila* embryos. Proc. RECOMB **8**: 157–66.

136 PETROVSKY, N. AND V. BRUSIC. 2002. Computational immunology: the coming of age. Immunol. Cell Biol. **80**: 248–54.

137 PRICE, T. S., R. REGAN, R. MOTT, et al. 2005. SW-ARRAY: a dynamic programming solution for the identification of copy-number changes in genomic DNA using array comparative genome hybridization data. Nucleic Acids Res. **33**: 3455–64.

138 PRUGNOLLE, F., A. MANICA, M. CHARPENTIER, J. F. GUEGAN, V. GUERNIER AND F. BALLOUX. 2005. Pathogen-driven selection and worldwide HLA class I diversity. Curr. Biol. **15**: 1022–7.

139 QUARANTA, V., A. M. WEAVER, P. T. CUMMINGS AND A. R. ANDERSON. 2005. Mathematical modeling of cancer: the future of prognosis and treatment. Clin. Chim. Acta **357**: 173–9.

140 RAHNENFÜHRER, J., N. BEERENWINKEL, W. A. SCHULZ, C. HARTMANN, A. VON DEIMLING, B. WULLICH AND T. LENGAUER. 2005. Estimating cancer survival and clinical outcome based on genetic tumor progression scores. Bioinformatics **21**: 2438–46.

141 RAKYAN, V. K., T. HILDMANN, K. L. NOVIK, et al. 2004. DNA methylation profiling of the human major histocompatibility complex: a pilot study for the human epigenome project. PLoS Biol. **2**: e405.

142 RAMAN, R., S. RAGURAM, G. VENKATARAMAN, J. C. PAULSON AND R. SASISEKHARAN. 2005. Glycomics: an integrated systems approach to structure-function relationships of glycans. Nat. Methods **2**: 817–24.

143 RAMMENSEE, H., J. BACHMANN, N. P. EMMERICH, O. A. BACHOR AND S. STEVANOVIC. 1999. SYFPEITHI: database for MHC ligands and peptide motifs. Immunogenetics **50**: 213–9.

144 REHMSMEIER, M., P. STEFFEN, M. HOCHSMANN AND R. GIEGERICH. 2004. Fast and effective prediction of microRNA/target duplexes. RNA **10**: 1507–1517.

145 RIBBA, B., K. MARRON, Z. AGUR, T. ALARCON AND P. K. MAINI. 2004. A mathematical model of Doxorubicin treatment efficacy for non-Hodgkin's lymphoma: investigation of the current protocol through theoretical modelling results. Bull. Math. Biol. **67**: 79–99.

146 RIBEIRO, C., V. PETIT AND M. AFFOLTER. 2003. Signaling systems, guided cell migration, and organogenesis: insights from genetic studies in *Drosophila*. Dev. Biol. **260**: 1–8.

147 ROBINSON, J., M. J. WALLER, P. STOEHR AND S. G. MARSH. 2005. IPD – the Immuno Polymorphism Database. Nucleic Acids Res. **33**: D523–6.

148 ROGNAN, D., A. STRYHN, L. FUGGER, S. LYNGBAEK, J. ENGBERG, P. S. ANDERSEN AND S. BUUS. 2000. Modeling the interactions of a peptide–major histocompatibility class I ligand with its receptors. I. Recognition by two alpha beta T cell receptors. J. Comput. Aided Mol. Des **14**: 53–69.

149 ROLLINS, R. A., F. HAGHIGHI, J. R. EDWARDS, R. DAS, M. Q. ZHANG, J.

JU AND T. H. BESTOR. 2005. Large-scale structure of genomic methylation patterns. Genome Res. **16**: 157–63.

150 ROST, B., J. LIU, R. NAIR, K. O. WRZESZCZYNSKI AND Y. OFRAN. 2003. Automatic prediction of protein function. Cell Mol. Life Sci. **60**: 2637–50.

151 SALI, A., R. GLAESER, T. EARNEST AND W. BAUMEISTER. 2003. From words to literature in structural proteomics. Nature **422**: 216–25.

152 SCHADT, E. E., S. W. EDWARDS, D. GUHATHAKURTA, et al. 2004. A comprehensive transcript index of the human genome generated using microarrays and computational approaches. Genome Biol. **5**: R73.

153 SCHERES, S. H., M. VALLE, R. NUNEZ, C. O. SORZANO, R. MARABINI, G. T. HERMAN AND J. M. CARAZO. 2005. Maximum-likelihood multi-reference refinement for electron microscopy images. J. Mol. Biol. **348**: 139–49.

154 SCHLESSINGER, A., Y. OFRAN, G. YACHDAV AND B. ROST. 2006. Epitome: database of structure-inferred antigenic epitopes. Nucleic Acids Res. **34**: D777–80.

155 SCHMIDT, K. A., C. V. HENKEL, G. ROZENBERG AND H. P. SPAINK. 2004. DNA computing using single-molecule hybridization detection. Nucleic Acids Res. **32**: 4962–8.

156 SETHUPATHY, P., B. CORDA AND A. G. HATZIGEORGIOU. 2005. TarBase: a comprehensive database of experimentally supported animal microRNA targets. RNA **12**: 192–7.

157 SEWER, A., N. PAUL, P. LANDGRAF, et al. 2005. Identification of clustered microRNAs using an *ab initio* prediction method. BMC Bioinformatics **6**: 267.

158 SHADEO, A. AND W. L. LAM. 2006. Comprehensive copy number profiles of breast cancer cell model genomes. Breast Cancer Res. **8**: R9.

159 SHAHI, P., S. LOUKIANIOUK, A. BOHNE-LANG, et al. 2006. Argonaute – a database for gene regulation by mammalian microRNAs. Nucleic Acids Res. **34**: D115–8.

160 SOREK, R., R. SHEMESH, Y. COHEN, O. BASECHESS, G. AST AND R. SHAMIR. 2004. A non-EST-based method for exon-skipping prediction. Genome Res. **14**: 1617–1623.

161 SPRINZAK, D. AND M. B. ELOWITZ. 2005. Reconstruction of genetic circuits. Nature **438**: 443–8.

162 STAMM, S., J.-J. RIETHOVEN, V. LE TEXIER, C. GOPALAKRISHNAN, V. KUMANDURI, Y. TANG, N. L. BARBOSA-MORAIS AND T. A. THANARAJ. 2006. ASD: a bioinformatics resource on alternative splicing. Nucleic Acids Res. **34**: D46–55.

163 STARK, A., J. BRENNECKE, R. B. RUSSELL AND S. M. COHEN. 2003. Identification of *Drosophila* microRNA targets. PLoS Biol. **1**: e60.

164 STEPHENS, D. J. AND V. J. ALLAN. 2003. Light microscopy techniques for live cell imaging. Science **300**: 82–6.

165 STETEFELD, J. AND M. A. RUEGG. 2005. Structural and functional diversity generated by alternative mRNA splicing. Trends Biochem. Sci. **30**: 515.

166 STEVANOVIC, S. 2005. Antigen processing is predictable: from genes to T cell epitopes. Transplant. Immunol. **14**: 171–4.

167 STOJANOVIC, M. N. AND D. STEFANOVIC. 2003. A deoxyribozyme-based molecular automaton. Nat. Biotechnol. **21**: 1069–74.

168 TOSELAND, C. P., D. J. CLAYTON, H. MCSPARRON, et al. 2005. AntiJen: a quantitative immunology database integrating functional, thermodynamic, kinetic, biophysical, and cellular data. Immunome Res. **1**: 4.

169 TSIAVALIARIS, G., S. FUJITA-BECKER AND D. J. MANSTEIN. 2004. Molecular engineering of a backwards-moving myosin motor. Nature **427**: 558–61.

170 TYSZKA, J. M., A. J. EWALD, J. B. WALLINGFORD AND S. E. FRASER. 2005. New tools for visualization and analysis of morphogenesis in spherical embryos. Dev. Dyn. **234**: 974–83.

171 VON DER LIETH, C. W., A. BOHNE-LANG, K. K. LOHMANN AND M. FRANK. 2004. Bioinformatics for glycomics: status, methods, requirements and perspectives. Brief Bioinform. **5**: 164–78.

172 WAN, S., P. V. COVENEY AND D. R. FLOWER. 2005. Molecular basis of peptide recognition by the TCR: affinity differences calculated using large scale computing. J. Immunol. **175**: 1715–23.

173 WANG, J., L. A. MEZA-ZEPEDA, S. H. KRESSE AND O. MYKLEBOST. 2004. M-CGH: analysing microarray-based CGH experiments. BMC Bioinformatics **5**: 74.

174 WANG, P., Y. KIM, J. POLLACK, B. NARASIMHAN AND R. TIBSHIRANI. 2005. A method for calling gains and losses in array CGH data. Biostatistics **6**: 45–58.

175 WANG, X., J. ZHANG, F. LI, J. GU, T. HE, X. ZHANG AND Y. LI. 2005. MicroRNA identification based on sequence and structure alignment. Bioinformatics **21**: 3610–4.

176 WEBER, M., J. J. DAVIES, D. WITTIG, E. J. OAKELEY, M. HAASE, W. L. LAM AND D. SCHUBELER. 2005. Chromosome-wide and promoter-specific analyses identify sites of differential DNA methylation in normal and transformed human cells. Nat. Genet. **37**: 853–62.

177 WEN, F., F. LI, H. XIA, X. LU, X. ZHANG AND Y. LI. 2004. The impact of very short alternative splicing on protein structures and functions in the human genome. Trends Genet. **20**: 232.

178 WILLIAMS, D. W. AND J. W. TRUMAN. 2005. Remodeling dendrites during insect metamorphosis. J. Neurobiol **64**: 24–33.

179 WONG, P. AND W. A. HOURY. 2004. Chaperone networks in bacteria: analysis of protein homeostasis in minimal cells. J. Struct. Biol. **146**: 79–89.

180 WOOD, W. AND A. JACINTO. 2005. Imaging cell movement during dorsal closure in *Drosophila* embryos. Methods Mol. Biol. **294**: 203–10.

181 XING, Y., A. RESCH AND C. LEE. 2004. The multiassembly problem: reconstructing multiple transcript isoforms from EST Fragment mixtures. Genome Res. **14**: 426–441.

182 YAMADA, Y., H. WATANABE, F. MIURA, et al. 2004. A comprehensive analysis of allelic methylation status of CpG islands on human chromosome 21q. Genome Res. **14**: 247–66.

183 YANG, Z. R. AND F. C. JOHNSON. 2005. Prediction of T-cell epitopes using biosupport vector machines. J. Chem. Inf. Model **45**: 1424–8.

184 ZHANG, H. AND M. LEVINE. 1999. Groucho and dCtBP mediate separate pathways of transcriptional repression in the *Drosophila* embryo. Proc. Natl Acad. Sci. USA **96**: 535–40.

185 ZHAO, Y., C. PINILLA, D. VALMORI, R. MARTIN AND R. SIMON. 2003. Application of support vector machines for T-cell epitopes prediction. Bioinformatics **19**: 1978–84.

186 ZHIHUA, L., W. YUZHANG, Z. BO, N. BING AND W. LI. 2004. Toward the quantitative prediction of T-cell epitopes: QSAR studies on peptides having affinity with the class I MHC molecular HLA-A*0201. J. Comput. Biol. **11**: 683–94.

Index

References to introductory descriptions of algorithmic and statistical bioinformatics methods and concepts are given in italic page numbers.

Bioinformatics - From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer

ISBN: 978-3-527-31278-8

Name Index

Bioinformatics – From Genomes to Therapies Vol. 3. Edited by Thomas Lengauer
Copyright © 2007 WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim
ISBN: 978-3-527-31278-8